IN 34
326
p-268

Thermocapillary Flow With Evaporation and Condensation and its Effect on Liquid Retention in Low-G Fluid Acquisition Devices

MSFC Center Director's Discretionary Fund
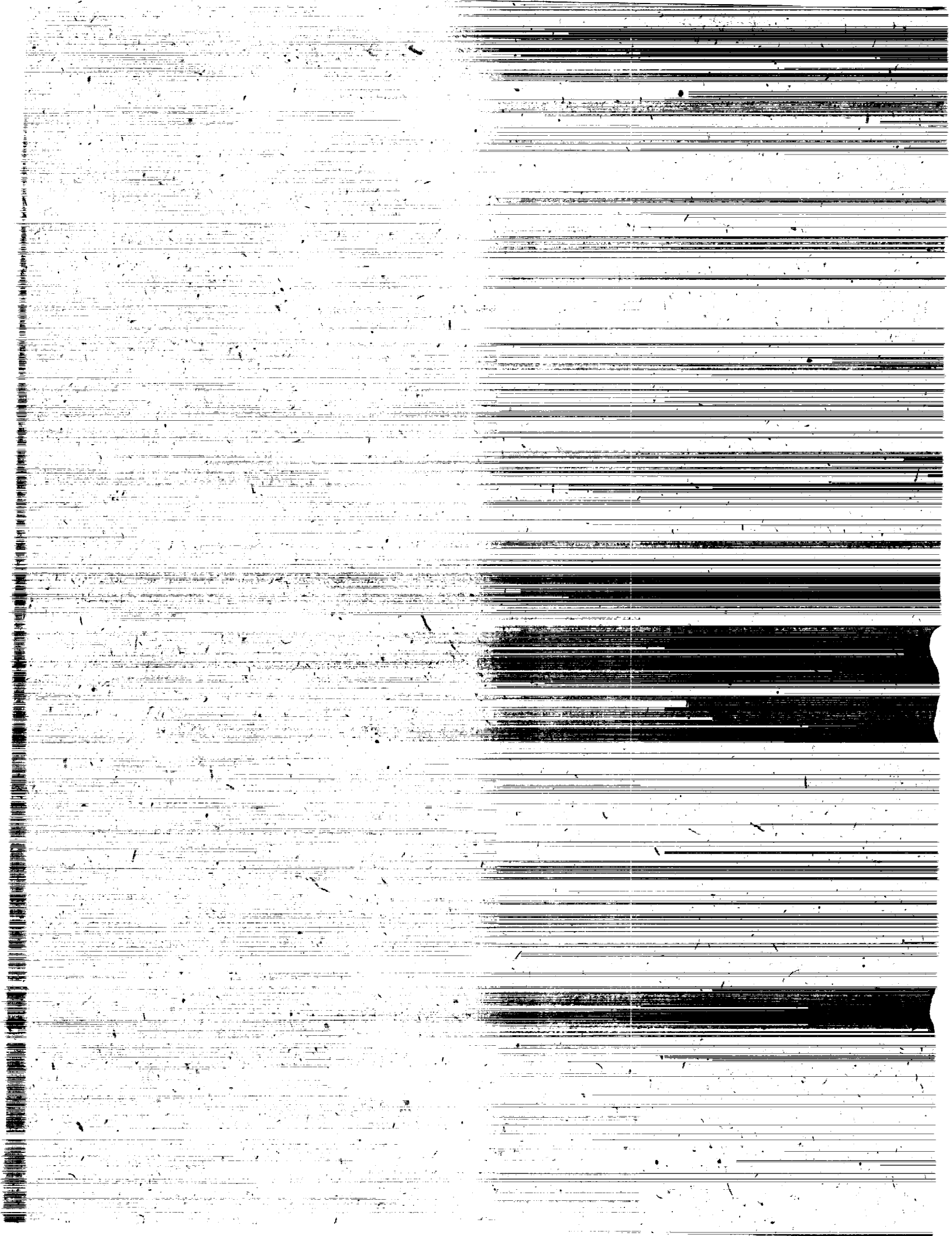
Final Report Project No. 91–15

G.R. Schmidt

NASA

Thermocapillary Flow With Evaporation and Condensation and its Effect on Liquid Retention in Low-G Fluid Acquisition Devices

G.R. Schmidt
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

ACKNOWLEDGMENTS

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

LIST OF SYMBOLS

Symbol	Definition
a_i	Earth-normalized acceleration ($i \equiv$ direction)
C	half-cavity circulation
cm	centimeter (10^{-2} meter)
D	characteristic dimension (pore width)
e	accommodation coefficient
F_i	thermocapillary stress force ($i \equiv$ direction)
f_μ	ratio of vapor to liquid viscosity
f_ρ	ratio of liquid to vapor density
g_i	Earth normal acceleration ($i \equiv$ direction)
g_0	Earth acceleration magnitude
h	heat transfer coefficient
j	mass flux normal to surface
J	total mass flowrate across meniscus (throughput)
k	thermal conductivity
K	degrees Kelvin
l	element surface length
L	latent heat
m	meter
M_W	molecular weight
n_i	unit vector normal to surface ($i \equiv$ direction)
P	pressure
P_d	dynamic pressure
P_h	hydrostatic pressure
P_l	momentum flux crossing surface (liquid)

P_v	momentum flux crossing surface (vapor)
q	heat flux normal to surface
Q	total heat rate across meniscus
R_g	universal gas constant
R_s	interfacial resistance
s	contour position
s_i	unit vector tangent to surface ($i \equiv$ direction)
t	time
T	temperature
T_{avg}	average surface temperature
T_h	maximum cavity temperature
T_l	minimum cavity temperature
T_0	vapor temperature
T_1	side wall temperature
V_i	velocity ($i \equiv$ direction)
W_β	test function ($\beta \equiv$ global node)
x_i	Cartesian coordinates
x',y'	coordinates meniscus-centered frame
$y^{(s)}$	meniscus surface x_2 position
$y_0^{(s)}$	meniscus x_2-coordinate centerline
α	contour angle of liquid surface
α,β,γ	global nodes
β	expansion coefficient
δ_{ij}	Kronecker delta
ΔP_b	bubble point pressure
$|\Delta T|$	subcooling/superheating temperature limit
γ	surface tension

Γ	surface/boundary area
φ	scaled temporal interval
ϑ_n	linear temporal interpolation function ($n \equiv$ time station)
κ	surface curvature
λ	thermal response ratio
μ	dynamic viscosity
μm	micron (10^{-6} meter)
ν	kinematic viscosity
ρ	density
σ_{ij}	stress tensor ($i \equiv$ direction, $j \equiv$ surface)
τ_{ij}	viscous stress tensor ($i \equiv$ direction, $j \equiv$ surface)
ζ	thermal diffusivity
ς	temporal parameter
ω	contact angle
Π	thermal potential
Π_β	pressure interpolation function ($\beta \equiv$ global node)
Φ_β	velocity/second-order interpolation function ($\beta \equiv$ global node)
Φ_β^*	velocity/second-order boundary test function ($\beta \equiv$ global node)
θ_β	temperature interpolation function ($\beta \equiv$ global node)
θ_β^*	temperature boundary test function ($\beta \equiv$ global node)
θ_i	modified stress ($i \equiv$ direction)
Θ_i	modified stress force ($i \equiv$ direction)
ξ_i	element naturalized coordinates ($i \equiv$ direction)
ξ, η	element naturalized coordinates
Ψ	stream function
Ψ_β	first order interpolation function
Ω	liquid volume

Dimensionless Parameters

Bi	Biot number
Bi_c	convective Biot number
Bi_{ke}	kinetic energy Biot number coefficient
Bi_m	mass transfer Biot number
Bo	Bond number
Bo_d	dynamic Bond number
Ca	Capillary number
Cr	Crispation number
Ma	Marangoni number
E	Evaporation number
Gr	Grashoff number
Oh	Ohnesorge number
Pr	Prandtl number
Ra	Rayleigh number
Re	surface tension Reynolds number
Sv	viscous stress parameter
Vr	recoil parameter
Λ	dimensionless latent heat

Subscripts

0	value at vapor temperature
1	value at side wall temperature
b	basic state
h	value at maximum cavity temperature
i	first-order tensor index
ij	second-order tensor index
$,i$	$\partial/\partial x_i$

$,ij$	$\partial^2/\partial x_i \partial x_j$
l	value at minimum cavity temperature

Superscripts

(e)	element
o	estimated value
(s)	surface
(v)	vapor
$'$	correction
$*$	dimensionless parameters

TECHNICAL PAPER

THERMOCAPILLARY FLOW WITH EVAPORATION AND CONDENSATION AND ITS EFFECT ON LIQUID RETENTION IN LOW-G FLUID ACQUISITION DEVICES

I. THE PROBLEM

A. Introduction

This study addresses a problem that directly pertains to the design of propellant and liquid management systems on spacecraft. In microgravity, liquid tends to assume an indeterminate orientation, and spacecraft tanks often incorporate devices that exploit surface tension to position, control, and maintain access to gas-free liquid. The outside surfaces of these liquid acquisition devices (LAD's) usually consist of a fine-mesh screen or porous barrier that segregates liquid from pressurant gas in the tank. Although such systems are used extensively with nonvolatile fluids, their performance with cryogens, particularly liquid hydrogen, is rather unpredictable. Prior tests have indicated that pressurization with heated vapor can reduce or even eliminate the ability of LAD's to retain gas-free liquid.

The objective of this study is to investigate the possible causes for retention loss in systems where the fluid is stored at or near its saturated state. The hypothesis that the convection processes arising in pores with dimensions and superheating/subcooling levels corresponding to cryogenic LAD applications are responsible for loss of retention will be evaluated. Although the study is directed at systems that maintain a pressure differential across a liquid and vapor/gas interface, understanding the nature of convection near the surfaces of liquid-filled wicking structures and porous media is important to the design of many engineering systems.

B. Background

An important aspect of fluid transfer in microgravity is the acquisition of gas-free liquid from storage tanks.[1][2] When subjected to an acceleration or gravitational field, liquid settles to the bottom of a tank to minimize its potential energy. In microgravity, however, the orientation is indeterminate and dictated by the competing effects of acceleration and surface tension. Oftentimes, the equilibrium position is not over the tank outlet, and special techniques and devices must be used to acquire liquid and to foster its expulsion.

One method is to apply scheduled thrusts via propulsive venting or attitude control rocket firings to provide a linear acceleration large enough to settle the liquid. After imposing an acceleration, the liquid or propellant can be fed to an engine or another receiver tank, depending on mission requirements. Although this method has been used successfully on upper stages, such as the Saturn V and Centaur, it tends to complicate flight operations and is impractical for missions involving refill from an onorbit fuel depot or transfer within a complex network of tanks.

Another method is to incorporate "passive" systems or devices that exploit surface tension to hold the liquid in a desired location or orientation.[3][4] These so-called LAD's have been used on spacecraft for years, primarily to control nonvolatile liquids and storable propellants. However, their convenience and effectiveness have made them appealing options for use with cryogenic propellants and fluids. LAD's can be used not only to guarantee access to vapor or gas-free liquid, but also to control center of gravity, damp sloshing motions, and facilitate venting of liquid-free pressurant.

Numerous LAD concepts have been proposed and used over the years. The most common ones are generally classified as either partial control devices or total communication systems. Partial control devices consist of a basket or trap that holds only a portion of the tank's liquid contents over an outlet while leaving the remainder free. An illustration of this concept installed within a spacecraft tank is shown in figure 1. The outer surface of these devices is ordinarily constructed of a fine-mesh screen or similar porous material. When the surface is exposed to vapor or gas, capillary forces in the screen pores prevent external gas from entering the LAD and mixing with the liquid. Alternatively, when liquid contacts the outside surface, the pores permit flow into and out of the device.
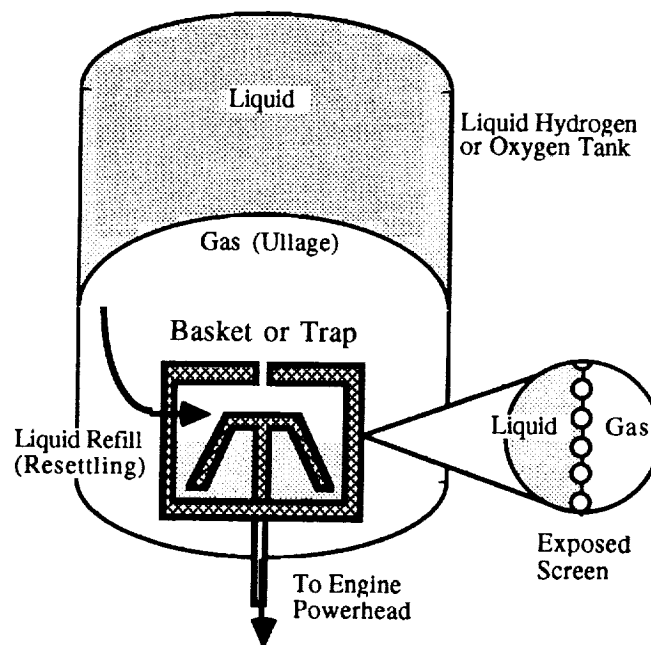
Figure 1. Partial control LAD.

These types of LAD's are primarily used to enable engine restarts in space or to provide continual access to propellant during high acceleration maneuvers. Upon application of thrust, the liquid settles and refills the trap which connects to the tank outlet. The trap is sized to provide enough gas-free propellant to restart and operate the engines until the remaining tank liquid collects over the outlet.

Total communication or gallery-type LAD's are designed to establish and maintain an uninterrupted flow path from the bulk liquid to the tank outlet. Figure 2 shows a schematic of a gallery LAD that would be used for onorbit resupply of propellant. Since liquid in microgravity tends to collect on tank walls, a common configuration includes a series of semicircular flow channels spaced only a short distance away from the wall. The wall-facing surface of each gallery consists of a

fine-mesh screen that permits the flow of liquid into the device, but inhibits the ingestion of gas. The galleries are all manifolded at the tank outlet, and as long as at least one of the channels remains in contact with the bulk liquid, tank pressurization will drive liquid through the channel into the tank outlet.
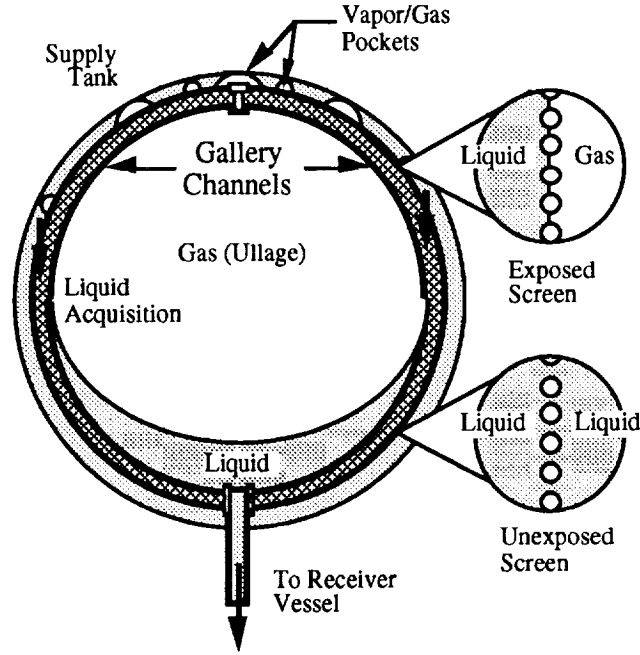
Figure 2. Total communication LAD.

Although fundamentally different in terms of design and function, both concepts rely on fine-mesh screens that allow liquid to pass into a channel or trap while inhibiting the flow of vapor. This unique behavior is due to the meniscus formed by surface tension between the screen mesh and liquid. As long as liquid adheres to the screen wires, vapor is kept out of the channel, and a pressure difference can be maintained to support flow through the device.

The most important design characteristic of a screen or porous LAD surface is the pressure differential it can maintain between the gas and liquid. The maximum possible differential is commonly referred to as the "bubble point pressure" and is a function of the screen weave and liquid properties. For a circular pore, the equation for bubble point pressure is easily derived from a force balance equating surface tension stress along the contact line to the pressure difference across the meniscus. In almost all applications, however, the screen consists of a complex interweave of wires or a random assortment of sintered metallic fragments that is extremely difficult to characterize analytically. Because of this, the equation for the screen's bubble point ΔP_b,[4] namely:

$$\Delta P_b = \frac{4\gamma \cos \omega}{D} , \tag{1}$$

is used to determine an effective hydraulic diameter D based on an experimentally measured pressure difference. For most fluids used in propellant applications, the contact angle ω is nearly zero and $\cos \omega = 1$. The size of commercial screens ranges considerably, and weaves with effective diameters as low as 1 μm to 10 μm are available.[5]

3

The performance of screen LAD's with nonvolatile liquids is well understood and has been demonstrated in numerous applications. When inside a screen channel or trap, these fluids are relatively immune to the minute departures in thermodynamic equilibrium caused by venting, pressurization, or conduction heating through the screen structure. Evaluating LAD retention performance is a straightforward problem that involves estimating screen size requirements from equation (1) and determining if the total pressure in the LAD ever results in a gas/liquid pressure differential that exceeds the screen's bubble point.

With cryogens and fluids stored in a saturated state, retention behavior is more complex. Since evaporation from screens exposed to gas or vapor is more prevalent than with nonvolatile liquids, wicking considerations become important.[4,5] Furthermore, because vapor may be generated within the LAD as a consequence of unavoidable heat leaks, features must be added to remove trapped vapor either through straight venting or periodic cooling and condensation.

Two very important aspects of spacecraft cryogenic fluid acquisition systems are tank pressurization and venting, which must be employed whenever the liquid is actively handled or controlled.[1] Pressurization is necessary to provide adequate back pressure for liquid expulsion to an engine powerhead or another tank. Venting, on the other hand, is performed during fill operations to relieve back pressure or is implemented periodically to compensate for liquid boiloff and vaporization. Both procedures involve changing the thermodynamic state of the pressurant relative to the liquid.

The most common pressurant gases used with nonvolatile liquids are ambient-temperature nitrogen and helium, which are supplied from a separate receptacle or tank. Since the liquid is subcooled, it is relatively immune to the changes in thermodynamic state caused by pressurization and venting. This type of pressurization system can also be used with cryogens and other saturated liquids, provided the pressurant has a lower boiling point than the liquid.

Another method, which has the potential for reducing cryogenic storage system weight and complexity, is "autogenous pressurization." With this concept, a portion of the liquid is drawn off, pressurized, heated, and reinjected into the tank as the pressurant gas. The benefits of this system are its elimination of an additional gas supply and its potential reduction in weight. The drawbacks are that the vapor and liquid in the acquisition system are thermodynamically linked, and the liquid is much more susceptible to heating conditions in the pressurant and LAD structure.

C. Statement of Problem

Prior experiments have shown retention of volatile liquids to be very sensitive to the type of pressurization system used. This is particularly true of liquid hydrogen (LH_2) which exhibits a notable drop in retention capability when pressurized using an autogenous system (i.e., heated hydrogen pressurant). Until recently, the reduction in screen bubble point observed in previous experiments was attributed to evaporation and dryout along the screen surface. This explanation would be plausible if the wicking rate had been too low to compensate for evaporation losses. However, when screen coupons and LAD's were subjected to highly evaporative environments, there appeared to be no significant degradation in retention capability. That is, application of direct conductive heating, pressurization with saturated hydrogen vapor, or pressurization with heated helium failed to incur much change in the nominal bubble point.[6-13]

4

The only time that a consistent reduction in retention occurred was when the tank and LAD surface were pressurized with heated hydrogen. Although no theoretically based investigation of this phenomena has been conducted, retention loss seems to ensue when the porous structure is subcooled with respect to the vapor. This fact, however, does not in itself constitute a complete or adequate explanation for retention loss. Rather, the cause is likely related to the liquid's response to thermodynamic nonequilibrium between the liquid and pressurant.

An immediate consequence of nonequilibrium is the convection arising from heat transport and temperature differences in the screen pores. The most likely convection modes in the size range considered here are thermocapillarity, which arises from thermal variation of surface tension along the meniscus, and evaporation and condensation, which become important if the liquid is near its saturated state. The flow arising from these convection modes, in turn, can influence surface morphology and may be responsible for the deterioration in retention observed in experiments.

The problem, therefore, is to determine whether the convection processes arising from thermodynamic nonequilibrium between the liquid and pressurant could lead to the retention failures observed in previous LAD experiments with liquid hydrogen.

D. Study Approach

The primary objectives of this study are to examine the retention problem and develop a plausible explanation for the cause of retention loss. In so doing, this study will improve understanding of the fluid flow associated with the subcooling and superheating of volatile, wetting liquids in small pores. Of particular interest is the manner in which these processes influence interfacial heat transfer and liquid surface morphology.

The study approach has been divided into six related tasks. The first is to formulate the problem in such a way that the hypothesis of convection about individual pore surfaces being the cause of retention loss can be evaluated. This requires development of a physical representation of the problem that models the transport phenomena and surface behavior of pores with length scales between 1 μm and 10^2 μm and superheat/subcooling levels of 10^{-1} K to 1 K.

In the second task, the governing equations and boundary conditions that characterize the convection modes associated with the physical model are derived. For the length and temperature levels considered here, the modes include thermocapillary stress, buoyancy, evaporation, and condensation. An important aspect of this task is to define a consistent set of dimensionless parameters that account for the fluid property and force ratios relevant to the problem. Following the approach of Burelbach et al.,[14] these parameters are obtained by scaling the governing equations and boundary conditions according to viscous scales. This approach is necessary for application of the one-sided approximation, which allows one to disregard transient vapor phenomena and to focus on liquid behavior. It also provides a basis for performing scaling comparisons, and renders the governing equations in a form more convenient for numerical analysis.

The third task involves development of a numerical model based on the finite element method. This model is incorporated in a computer code that calculates steady-state velocity, pressure, and temperature distributions, while accounting for deformation of the free surface. A major challenge in modeling the highly wetting fluids considered in this problem is the handling of low contact angles in the meniscus interline region. The large surface derivatives near these points promote substantial

5

gradients in the flow parameters which are best handled using higher-order interpolation functions. Another challenge is the large number of terms that must be included in the calculation of surface geometry. The equation used for this is derived from the normal stress balance and is comprised of terms representing pressure, thermally induced surface tension variation, momentum change between the vapor and liquid, and viscous stress.

In the fourth part of the study, the dimensionless parameters derived from scaling the governing equations and boundary conditions are used to assess the relative influence of different terms in the equations for velocity/pressure, temperature, and surface geometry. This provides an approximate indication of which forces and convection mechanisms predominate at different length and temperature scales. Such a scaling analysis is also valuable for estimating physically relevant parameter values.

The fifth task involves numerical examination of thermocapillary and interfacial flow about a fixed, low contact angle surface. By assuming vanishingly small Bond, Capillary, and Crispation numbers, one can safely ignore deformation and assume a circular geometry. The purpose here is to examine the separate and combined effects of thermocapillarity, condensation/evaporation, and contact angle on interfacial temperature and pressure, heat transfer, circulation, and surface stress. It is important to assess the fundamental behavior of convection since it directly influences the interfacial characteristics responsible for surface instability and loss of retention.

The sixth and final task involves examination of surface deformation. The criteria that link surface deformation to retention loss are developed, and an analysis of first-order effects is performed by assuming decoupled solutions for the flowfield and meniscus. Meniscus geometry is recalculated using the steady-state interfacial velocity, pressure, and temperature distributions determined in the fifth task. The purpose is to evaluate the relative influence of deformation-producing terms and dimensionless parameters in the equation for surface curvature. Finally, the simultaneous solution of the steady-state flowfield and surface is considered. The intent is to identify retention loss mechanisms that could arise as a result of either thermocapillary or interfacial convection.

E. Significance of Study

The significance of this study lies in both its contribution to engineering applications involving heat transfer in porous media and its advancement to the understanding of two-phase microgravity fluid behavior. Although the emphasis is on retention in cryogenic liquid acquisition systems, the results and methodology have application to a variety of fields. From an applications standpoint, the study addresses a problem that is particularly relevant to devices that encounter condensation and evaporation along a very fine porous structure. Heat transfer and overall performance of such systems are strongly influenced by the combined-mode convection associated with thermocapillarity, evaporation, and condensation.

A good example of this is the heat pipe,[15] where prediction of latent heat transport across the evaporator and condenser wicking surfaces is crucial for determining the total axial heat transfer rate of the device. In the evaporator section, thermocapillarity and evaporation both tend to convect superheated liquid onto the surface and promote transfer of latent heat and mass into the vapor. In the condenser, however, thermocapillarity and condensation counteract one another in that thermocapillarity tends to cool the interface and augment heat transfer, while condensation restricts

it by raising interfacial temperature. The combination of these modes complicate the balance between heat and mass transfer at different ends of the pipe which can either augment or inhibit the overall heat transfer capability of the system.

Studying the influence of interfacial convection on heat transfer and retention in small liquid cavities also represents an original contribution to the field of microgravity fluid mechanics. The problem considered in this study can be viewed as a subset of a broader class of problems addressing free liquid surfaces subjected to the combined effects of thermocapillary stress and other convection mechanisms. Although many problems of this type have been studied before, most work in this area to date has concentrated on the separate or simultaneous action of thermocapillarity and buoyancy—the most familiar cases being the liquid-bridge and float-zone configurations analyzed in materials processing.[16] These regimes are particularly useful for modeling and predicting the melt concentrations in various ground-based crystal growth techniques. They are also often used to model low-gravity containerless processing due to the unavoidable presence of extraneous accelerations and disturbances. In either case, buoyancy must be accounted for because of the large length scales associated with the problem (~1 cm to 1 m). Furthermore, since the ratio of viscous forces to surface tension is usually negligible, free surface deformation is dictated almost entirely by hydrostatic effects and is independent of flow conditions.

F. Summary

The goal of this study is to improve the understanding of convection processes associated with the subcooling and superheating of volatile, wetting liquids in a microgravity environment. Although the results are pertinent to low-gravity applications, the selection of length and temperature scales is geared more toward systems having small pores and liquid passages. The primary reason for conducting such a study is to determine if flow effects around these surfaces could lead to conditions of retention failure when subjected to different environmental conditions.

The study approach consists of the following: (1) formulation of a physical model; (2) derivation of the governing equations, boundary conditions, and dimensionless parameters that characterize fluid flow with thermocapillary stress, buoyancy, evaporation, and condensation; (3) development of a numerical model that calculates the steady-state velocity, pressure, and temperature distributions in the pore, while accounting for deformation of the free surface; (4) scaling analyses to assess the relative influence of different terms in the equations for velocity/pressure, temperature, and surface geometry; (5) numerical examination of thermocapillary and interfacial flow about a fixed, low contact angle surface; and (6) examination of surface deformation via both decoupled and integrated solutions.

II. REVIEW OF LITERATURE

A. Introduction

The approach draws on work from several different research areas. In this section, literature from each field is reviewed to develop an integrated basis for analyzing the problem. First, the experimental work with liquid hydrogen and various pressurant conditions that led to identification of the "retention problem" is presented. This section summarizes the results of tests in which

screened coupons and LAD channels were subjected to different pressurant types, temperatures, and conductive heat loads.

Since the problem involves convection on a microscopic scale, the literature was reviewed primarily in the areas of thermocapillarity, evaporation, and condensation. When it comes to characterizing thermocapillary flow, the situation examined here is very similar to the liquid bridge and float-zone configurations studied in materials processing. However, important differences, such as the presence of liquid/vapor phase change, small contact angles, and mass transfer with a relatively large liquid reservoir, make the problem addressed here unique. It is also expected that because of the small length scales, viscous and flow-related phenomena could compete with surface tension and cause appreciable surface deformation under some conditions. Much of the interfacial physics and assumptions regarding surface equilibrium also appear in prior investigations of thin film and capillary jet instability. This study, however, considers a pore geometry and dimensional range in which intermolecular attraction between the liquid and solid is best accounted for by a contact angle constraint rather than a corrected surface or body force. By ignoring intermolecular and, as will be shown later, acceleration body forces, it is assumed that any deformation of the free surface is caused by surface tension variation, pressure, interfacial momentum flux, and/or viscous stress.

Note that several dimensionless groupings are referred to throughout this section and remaining portions of the text. Most of these parameters are derived in section III as part of the theoretical development, but a few are taken directly from the references. A definition of all groupings is provided in table 2.

B. Liquid Retention

Several investigations of LAD retention performance with cryogenic liquids have been performed since the early 1970's. Among the earliest was Burge and Blackmon[6] who tested the retention capabilities of fine-mesh screen samples in LH_2 while subjected to warm hydrogen pressurant. In their experiments, hydrogen vapor was evaporated from a screen and blown back through a fan and heater. Contrary to their earlier experiments with nitrogen, they found that warming the hydrogen vapor by only 5 to 10 K above the bulk liquid temperature caused premature retention loss (i.e., lowering of the effective bubble point pressure). For vapor temperatures above 30 K, the reduction in liquid retention reached a maximum of ~70 percent of the nominal bubble point pressure.

Burge and Blackmon later conducted another experiment in which screen samples were tested in an apparatus that enabled tighter control of pressurization rate and heat flux. Although the heater was placed closely above the screen, there was no fan or associated forced convection as in the first test. Results showed no premature retention loss for vapor temperatures up to 40 K. This behavior, however, was not observed in a third larger-scale experiment in which a screen basket was submerged in LH_2 and then raised, while inverted, into hydrogen vapor of various temperatures. With temperatures of 40 K, retention loss was clearly evident and caused an immediate draining of the basket, but with temperatures less than 30 K, the basket retained the liquid until the hydrogen gas pressure was allowed to drop below the vapor pressure of the liquid.

Blackmon[7] later tested a vertically oriented screen channel in which LH_2 was withdrawn up through the device and out the top of the tank (i.e., minus one-g expulsion). He tested with both hydrogen and helium gas, and found that warm pressurant, regardless of gas type, resulted in

premature retention loss. The reduction in retention was as much as 70 percent of the design bubble point with hydrogen vapor, but was only about 20 percent with helium. In some of the tests with warm pressurant, heat conduction into the channel through fittings and solid portions was sufficient to cause vaporization of the liquid and to eliminate retention capability altogether. Tests with colder pressurant close to the hydrogen saturation temperature yielded generally better than predicted results and no premature screen breakdown. There was, however, more variability in retention capability with hydrogen than with helium.

Using the same apparatus employed by Burge and Blackmon in their second series of screen sample tests, Cady[8] later measured the bubble points of different screens in LH_2 while subjecting them to various calibrated heating rates. He found that, even at rates greatly exceeding the worst case for a spacecraft application, the maximum reduction in bubble point was less than 15 percent—a result consistent with the less extensive testing by Burge and Blackmon. Wicking through the screen and evaporation at the surface effectively absorbed the heat. This was in spite of the test conditions in several of the runs being deliberately established so that the liquid beneath the screen was superheated with respect to the local liquid pressure.

Paynter and Page[9] and Warren[10][11] conducted minus one-g expulsion tests with LH_2 in a 63.5-cm diameter spherical tank fitted with a total communication acquisition device comprised of eight screened channels. In tests where the outflow was continuous, no screen breakdown occurred, regardless of the pressurant type (i.e., hydrogen or helium) or pressurant temperature. However, in tests where the expulsion was performed intermittently, with alternating hold and outflow periods, retention loss occurred repeatedly with warm hydrogen pressurant but not with warm helium. The pressure difference across the screen imposed by hydrostatic head and screen flow-through loss, although not measured, was probably not more than 80 percent of the bubble point of the screens used in the LAD. Warren concluded that either hydrogen or helium may be used as pressurant as long as there are no stagnant regions within the LAD.

Bennett[12] also conducted retention experiments involving a 34-cm high partial control LAD (i.e., start basket) in LH_2. At the beginning of each test run, the basket was submerged in liquid. The liquid was then drained to a level below the basket and held there so that the basket was completely surrounded by the pressurant gas. Tests were conducted with both gaseous hydrogen and helium. In all hydrogen pressurant tests, the vapor was cooled to about 90 K prior to injection into the test dewar, and the screen failed even before the liquid level was lowered to the bottom of the basket. On average, the hydrostatic head at the point of failure was 50 percent of the normal bubble point of the screen in LH_2. With helium pressurant, the liquid level could always be lowered below the basket, and subsequent retention was demonstrated for periods up to 3 h.

The most recent and possibly comprehensive investigation of LH_2 retention was performed by Meserole and Jones,[13] who measured the breakdown pressure of a semicircular screened channel with different pressurant types (i.e., helium, ambient temperature hydrogen, and parahydrogen), pressurant temperatures, and degrees of liquid stagnation. With helium, no measurable variation in retention performance was observed, and the breakdown pressure in every test was close to the predicted screen bubble point. With hydrogen vapor, however, retention exhibited strong sensitivity to pressurant temperature. Cool vapor at approximately saturation temperature yielded breakdown pressures close to the predictions, which were slightly less than with helium. With a warm vapor, the breakdown pressure was significantly reduced, and temperatures as low as 15 to 20 K above saturation caused sharp reductions in liquid retention capability.

Meserole and Jones also found that interrupting the outflow and stagnating liquid in the LAD with hydrogen pressurant markedly reduced retention capability, even with vapor temperatures a few degrees above the bulk liquid temperature. Retention failure repeatedly occurred shortly after the outflow was interrupted, when the differential pressure was just 20 percent of the nominal bubble point.

Meserole and Jones proposed several explanations for their observations and those made by previous researchers. They felt that evaporative cooling was responsible for the strong tolerance to heated hydrogen vapor observed with individual screen samples, while the reduced retention was attributed to condensation of warm vapor on the screen. During liquid expulsion, the vapor is pressurized, and the screen pores become subcooled with respect to the vapor, thereby promoting condensation. However, in screen sample experiments, either there is no active pressurization or the screen is heated directly. Thus, the pores are superheated with respect to the vapor, and evaporation occurs. With helium, evaporation always takes place regardless of whether the vapor is pressurized or not.

Meserole and Jones felt that the results of these tests helped clarify the differences observed with various pressurant conditions. Although they did not offer an explanation as to why evaporation and condensation influence retention so differently, they felt that further investigation was warranted to determine how screen heating and cooling alters the attachment of the liquid surface to the screen wires and causes premature breakdown.

C. Thermocapillary-Driven Convection

Most prior work involving thermocapillarity and its closely associated topic of combined thermocapillary-buoyancy driven flow has been done in the area of materials processing.[16] Specific applications that have served as an impetus for investigations are crystal growth,[17] glass manufacturing,[18] and industrial processes such as welding.[19] Although much of this research, particularly in conjunction with the study of flowfields in containerless processing, is relevant to the study of convection-induced retention loss, direct application is limited by the fact that the melts associated with materials research typically have larger contact angles ($\omega \sim 90°$) and do not involve appreciable liquid-vapor phase change.

A popular geometry for the study of pure and combined convection is the rectangular cavity with isothermal side walls and a horizontal upper free surface. Numerical investigations by Bergman and Ramadhyani,[20] Bergman and Keller,[21] Jue et al.,[22] and Hadid and Roux[23] yielded insight into the sensitivity of flow variables to key dimensionless parameters. In all cases, the variation of free surface geometry was treated as a lower-order effect and neglected by assuming a small thermal variation of surface tension with respect to a reference (i.e., low Crispation number Cr) and low ratio of viscous to surface tension forces (i.e., low Capillary number Ca).

Several researchers have applied the same geometry to evaluate thermocapillary convection with a deformable liquid surface. Sen and Davis,[24] Sen,[25] and Strani et al.[26] applied asymptotic methods to determine the flowfield and free surface shape at the ends and core of differentially heated slots. They found that thermocapillary convection with either fixed contact angle or pinned end conditions causes bulging in colder surface regions, where the flow accumulates and increases pressure. Zebib et al.[27] and Carpenter and Homsy[28] examined interface geometry by means of perturbation techniques, and Sen[25] applied an independent coupling of the meniscus force balance

10

condition to show that the interface can undergo significant deflection with large values of Cr. Assuming a prescribed contact angle, Cuvelier and Driessen[29] found that the free boundary is highly sensitive to Ca.

Cylindrical geometries have also been used to investigate thermocapillary and buoyancy-driven flow in melt regions. Earlier studies by Fu and Ostrach,[30] Shen et al.,[31] and Kobayashi[32] assumed a fixed surface independent of convective effects. Xu and Davis[33] studied thermocapillary instabilities of a cylindrical column. Somewhat more recently, Duranceau and Brown,[34] Hyer et al.,[35] Lan and Kou,[36] and Zhang and Alexander[37] addressed the problem of a cylindrical float zone with a deformable interface. They applied a pinning condition at the contact point and observed little flow induced deformation. The notable deformation in the buoyancy-dominant one-g regime was attributed to hydrostatic effects, while the surface remained essentially flat in a microgravity environment. These results and the assumption of vanishingly small Cr and Ca were entirely consistent with the relatively large domains considered in these studies. However, Kamotani and Platt[38] recently showed that variation in surface geometry could greatly influence convection within the cavity. Through experiments and numerical analysis, they compared the convection and heat transfer characteristics of a flat and curved 10°-contact angle surface, and noted a marked reduction in thermocapillarity, flow intensity, and local heat transfer rate with the highly curved surface.

All of these results indicate that convection-induced deformation can be neglected in float-zones and macroscopic free surface flows, where Ca and Cr are ordinarily quite small. The length-scales of the retention problem, however, are large enough that the values of Ca and Cr fall within the range where deformation can be significant. When viewed in this context, the results suggest that retention loss may be related to convection-induced deformation of the free surface.

D. Liquid Evaporation and Condensation

Most past studies of evaporating menisci have focused on either the intrinsic meniscus, which is the portion of the interface characterized solely by surface tension forces, or the submicron region near the interline where the solid substrate plays an important role in surface phenomena. Work dealing with the submicron region has concentrated on determining the heat transfer and stability of thin liquid films and menisci close to solid surfaces. In this region, buoyancy forces and convective terms in the momentum equation are typically ignored, and emphasis is placed on the relationship between thermocapillary stress and intermolecular forces between the contact surface and liquid (see Potash and Wayner,[39] Renk and Wayner,[40] Werhle and Voulelikas,[41] and Mirsamoghadam and Catton[42]). In a study of the stability of an evaporating liquid film, Burelbach et al.[14] treated the van der Waals forces as a body-force term in the momentum equation. They noted that the combined action of this attraction, thermocapillarity, and evaporation can lead to significant instability and deformation of the surface. This behavior had been observed before in other references cited by Burelbach et al.

Swanson and Herdt[43] developed an analytical model of an evaporating meniscus in a capillary tube accounting for all regions of the meniscus. Their assumptions typified those of prior studies; they ignored hydrostatic pressure variation and convective terms in the momentum equation, and accounted for Marangoni effects only in the thin film region. The solution for the thin film near the wall was subsequently coupled with that through the center of the tube. Swanson and Peterson[44] developed a mathematical model of the intrinsic meniscus in a V-shaped channel for an unsaturated

wetting liquid evaporating into air. The formulation of the model generated matching (or coupling) parameters that accounted for phenomena in the interline region.

In a study very similar to the problem in this report, Chen et al.[45] applied a one-sided model of a superheated liquid to investigate motion in a square cavity with isothermal side walls. They determined the flowfields and interface heat transfer rates at moderate values of Marangoni number Ma, interfacial resistance Rs, and Biot number Bi. Buoyancy forces were neglected, and meniscus geometry was calculated independently based on hydrostatic pressure considerations alone. This assumption may be invalid for an evaporating surface. In this case, temperature-dependent vapor recoil arises in the normal force balance and may influence geometry even at low Ca and Cr.

E. Summary

In this section, literature relevant to the retention problem was reviewed. It first addressed prior tests of screen retention capability with liquid hydrogen and discussed the difficulties of pressurizing with heated hydrogen vapor. Although retention failure was repeatedly encountered in several experiments, no plausible explanation of its cause had been proposed. Meserole and Jones[13] improved understanding in this area by distinguishing the probable surface convection modes associated with vapor and inert gas pressurization. However, they did not suggest reasons why the mode resulting from vapor pressurization (i.e., condensation) contributed to retention loss.

Since retention loss was observed in pores having characteristic dimensions of ~1 μm to 10^2 μm, literature on the convection modes appropriate to these length-scales, namely thermocapillarity, evaporation, and condensation, was surveyed. Most of the work reviewed in the area of thermocapillary flow dealt with variants of the well-known float-zone and liquid bridge problems. Although the surface pore problem is similar to these, it is unique in several ways: (1) the liquid is volatile such that fluid motion results from the simultaneous action of interfacial mass transfer (i.e., evaporation or condensation) and thermocapillary stress; (2) buoyancy (whether in one or zero-g) is negligible due to the small dimensions of the problem domain; (3) the interface curvature (in the isothermal state) is nearly constant due to vanishingly small Bond numbers; (4) the liquid is wetting and intercepts the solid boundaries at a constant acute contact angle; (5) the pore connects to a large reservoir that permits balancing of mass flow through the pore; and (6) the ratio of viscous forces to surface tension is large enough to cause appreciable surface deformation under some flow conditions.

Prior studies in thermocapillary flow were useful for understanding basic flow patterns and parameter sensitivities, but none fully addressed the issues associated with the retention problem. The same held true for research in the field of interfacial flow and heat transfer which is primarily geared toward stability of thin films and meniscus interline regions. In addition, the length scales of interest were typically much smaller and, therefore, required inclusion of solid/liquid intermolecular forces. However, the approach used for modeling interfacial equilibrium and its influence on mass transfer is nevertheless applicable to the problem in this study.

12

III. THEORETICAL MODEL DEVELOPMENT

A. Introduction

The problem is to evaluate the influence of thermocapillarity, evaporation, and condensation on fluid flow, heat transfer, and retention in a small pore of volatile, wetting liquid exposed to its own vapor. To analyze this problem, one must first define a physical reference and corresponding mathematical model that characterize the thermophysics and imposed conditions on the surfaces of porous media. The challenge is to capture all of the key aspects related to convection and free surface behavior, while keeping the model simple enough for straightforward scaling and numerical analyses.

B. Physical Model

Since the retention problem has not been theoretically examined before, definition of the physical model is one of the primary contributions of this study. To begin, one must identify the features of this problem that distinguish it from the traditional approach of evaluating pore retention (equation (1)). One feature that arises from imposed pressurization and/or direct heating of the LAD screen is a departure from thermodynamic equilibrium between the liquid and pressurant. It is important that the physical model adequately represents this deviation and properly portrays the relative quasi-equilibrium thermodynamic states between the pore structure, liquid, and pressurant gas. A second feature is the complex geometry associated with screens and porous media. The traditional approach of estimating screen retention capability (equation (1)) is based on a circular pore geometry that permits application of a simple force balance. Likewise, in developing a physical model for this problem, it is advantageous to define a domain whose boundaries are not only easy to characterize analytically but also produce the convection modes relevant to the retention problem.

To define appropriate thermodynamic relationships between the liquid, gas, and pore boundaries, the nonequilibrium conditions established in the retention tests described in section II are reviewed. The types of heating conditions that the LAD's and screen samples were subjected to can be generalized as follows: (1) pressurization with heated vapor,[6 7 9–13] (2) evaporation with an overhead heater,[6 8] (3) evaporation with an overhead heater followed by recirculation over screen surface,[6] (4) evaporation through conductive heating of screen,[8] and (5) pressurization with heated helium.[7 9–11 13]

In case (1), vapor is introduced into a vessel containing an LAD at a higher pressure and temperature than the saturated liquid in the channel or trap. Upon pressurization, the liquid is no longer in a saturated state, and it becomes subcooled with respect to the vapor. The process in case (1) incurs the most retention failures and represents a subcooling of the liquid and screen relative to a saturated or superheated vapor. In cases (2) and (3), the vapor is obtained through evaporation, and liquid in the vicinity of the surface is superheated with respect to the surrounding vapor. In case (2), the pressurant is not actively recirculated over the screen, and no significant retention loss is observed. But in case (3), recirculation takes place and retention failures are more numerous.

In case (4), liquid is evaporated by heating through the screen wires. As in case (2), the liquid becomes superheated relative to the vapor. However, because the entire wire is heated, the region of superheat is expected to extend well around the pore vicinity. This case is also similar to

case (2) in that no significant drop in retention capability occurs. In case (5), heated helium is introduced at a higher pressure and temperature at the same general conditions as case (1). Although the liquid is again subcooled with respect to a higher pressure and temperature pressurant, no reduction in retention capability is observed. The main aspect that distinguishes cases (1) and (5) is the higher level of nonequilibrium between the liquid and gas in the presence of helium, which serves to reduce both evaporation and condensation.

The relative thermodynamic states in these tests represent either a superheating or subcooling of the liquid and wire structure relative to a saturated vapor. In addition, the presence of helium gas can be viewed as an imposed reduction in vapor/liquid equilibrium along the exposed surface.

Another important aspect of the physical model is the geometry of the problem domain and free surface. As stated in section I.C, retention loss is likely due to instability or adverse deformation of the meniscus in response to the convection ensuing from thermodynamic non-equilibrium. Therefore, the free surface must be defined within the context of a control volume to enable application of governing equations and boundary conditions for fluid motion and heat transfer. This is difficult because the contact surface of a typical porous structure usually consists of a complex interweave of metal wires or random arrangement of sintered metallic fragments. Thus, basing an assessment on an actual application would be unduly complicated by the variability in contact surface geometry and uncertainty regarding interior flow conditions.

The problem is greatly simplified by assuming a two-dimensional domain that approximates the geometrical characteristics of a porous structure. A domain that was considered early in the study is depicted in figure 3. It basically represents a two-dimensional cross section of a screen with individual menisci attached to each wire. An advantage of this configuration is that the curved side walls closely model the shape of screen wire surfaces and properly account for the surface's ability to readjust to pressure changes by moving along the wires.

For an initial investigation, however, the domain in figure 3 introduces many complexities that would have to be accommodated in a sophisticated numerical model and procedure. One of the most significant is the lack of velocity, pressure, and temperature data for the boundaries defining the liquid channel. An additional determination of bulk velocity, pressure, and temperature in the channel would be required prior to solving the flowfield around the surface and wires. Secondly, modeling a moving deformable meniscus would add considerable complexity to the numerics of the problem.
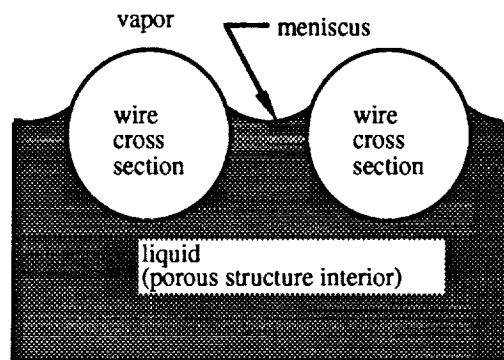
Figure 3. Simplified pore schematic.

The model is further simplified by making the following assumptions. First, at the point of retention loss for a static isothermal fluid, the meniscus acquires a maximum curvature, and its contact points lie on a chord connecting the centers of the circles in figure 3. For the nonequilibrium case, the meniscus should assume the same position relative to the screen wires at a pressure difference corresponding to the reduced bubble point. The approach for analyzing the retention problem then is to determine whether a stable, steady-state solution of the flowfield and surface exists at the minimum separation distance. This assumption allows one to fix the endpoints at this position and ignore movement of the meniscus boundary along the wire surface.

A second assumption is that the convection responsible for retention loss occurs close to the surface, and is relatively independent of flow below the wires and in the LAD channel. Each pore is so small relative to the LAD channel that the influence of bulk flow should be negligible. By assuming fixed contact points and restricting flow effects to the surface vicinity, the curved side walls are probably nothing more than a second-order effect on convection within the pore.

Based on these assumptions, one resorts to the greatly simplified domain depicted in figure 4, which is very similar to the geometry examined by Chen et al.[45] It consists of a two-dimensional rectangular groove partially filled with an incompressible, Newtonian liquid. Although it will be shown later that gravitational effects can be ignored, for the sake of completeness, a uniform acceleration field is assumed to point downward in the $-x_2$ $(-y)$-direction. The left and right sides of the pore (boundaries 1 and 3, respectively) consist of vertical side walls. The lower boundary (2) opens to a large reservoir to enable balancing of liquid flow through the cavity and modeling of capillary structure performance.
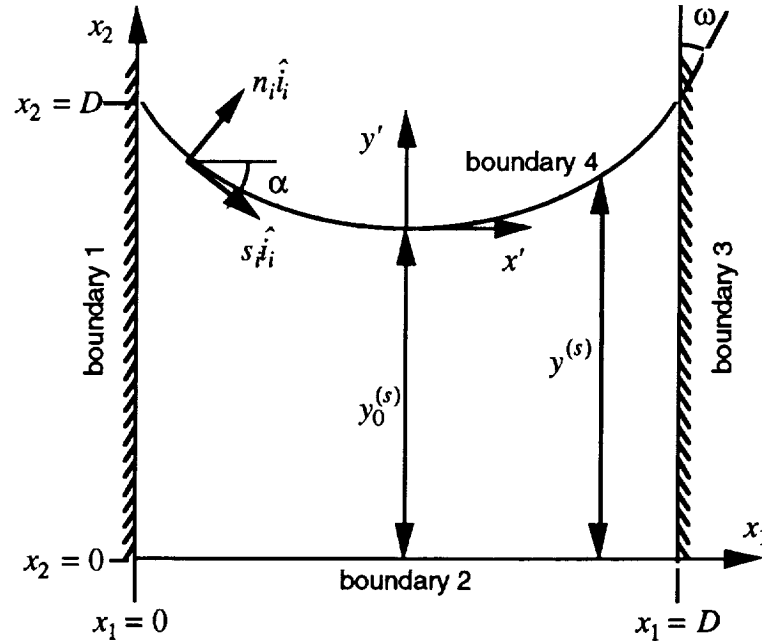
Figure 4. Problem domain.

The upper surface (boundary 4) is represented by a curved meniscus which is symmetric about the pore center-line and bounded by an inert vapor. The shape of the interface is defined by the function $y^{(s)} = y^{(s)}(x_1,t)$ where the surface height above the base $y^{(s)}$ is a function of the lateral

coordinate x_1 and time t. The interface is assumed to wet the solid side walls at an acute contact angle ω, and is further characterized by the unit normal and tangent vectors (n_i and s_i) and contour angle α.

Compared to the wire-based geometry in figure 3, the domain in figure 4 implicitly represents a case where the wire diameter is much greater than the separation distance. In addition, fixing the contact point (i.e., $x_2(0) = D$ and $x_2(D) = D$) is akin to assuming that the meniscus can no longer adjust to increased pressure differences between the vapor and liquid, and that it is at the maximum bubble point pressure.

The configuration in figure 4 is similar geometrically to the domain studied in float-zone and liquid bridge problems.[34-37] However, there are several differences in the treatment of velocity boundary conditions that make this problem unique. On boundaries 1 and 3, for instance, the x_1 and x_2 components of velocity are held at zero, $V_i = 0$, by invoking impermeability and no-slip. Along the meniscus, Neumann and Dirichlet-type boundary conditions are accounted for simultaneously by applying the x_1-direction thermocapillary stress condition to V_1, and calculating V_2 directly from the estimate of V_1 and the normal velocity $V_i n_i$ arising from interfacial nonequilibrium. On boundary 2, it is assumed that the flow into and out of the cavity is parallel and uniform. Consequently, $V_1 = 0$, and the average flux required to balance the total flow across boundary 4 and adjust for surface deformation is applied uniformly as the V_2 boundary condition.

The temperature boundary conditions are basically the same as a float-zone. The side walls and vapor are assumed to be isothermal and are held at temperatures of T_1 and T_0, respectively. T_0 represents the fluid's saturation temperature, while T_1 can either be lower or higher depending on whether the pore boundaries are subcooled $T_1 < T_0$ or superheated $T_1 > T_0$ with respect to the vapor. The boundary condition for temperature on the meniscus is handled using a Robin-type condition which will be discussed in section III.C. Unlike the float-zone problem, the temperature along boundary 2 is held constant at the side wall temperature T_1. Thus, it is assumed that the liquid residing in the interior capillary structure is either superheated or subcooled with respect to its surrounding vapor.

Some of the other major assumptions include incompressibility, Newtonian fluid, one-sided approximation for all vapor properties other than density, linear equation of state for density and surface tension in terms of temperature, and restriction of liquid/vapor phase change to free liquid surface.

C. Mathematical Model

The equations needed to solve the five dependent variables describing this system, i.e., V_i, P, T, and $y^{(s)}$, are obtained from the imposed conditions on boundaries 1, 2, and 3, the governing equations of fluid motion, and the jump conditions across boundary 4. The calculation of $y^{(s)}$ is indirectly linked to solution of the other variables and is described more extensively in section IV. To summarize, a surface is determined that satisfies the constant contact angle constraint and normal stress condition involving interfacial pressure, temperature, and velocity.

Throughout the remaining sections, this report makes use of linear equations of state for surface tension and density. These approximations are all based on T_1 which represents the minimum

16

temperature in the liquid and vapor domain, and depends on the heating mode applied. With superheating, T_l is the temperature of the saturated vapor $T_l = T_0$, while for subcooling, it is the boundary temperature $T_l = T_1$.

The governing relationships for velocity, pressure, and temperature are based on constant thermophysical properties and obtained from the continuity, momentum, and energy equations for an incompressible fluid,[47] namely:

$$V_{i,i} = 0 \ , \tag{2}$$

$$\rho \frac{\partial V_i}{\partial t} + \rho V_j V_{i,j} + P_{d,i} - \tau_{ij,j} - \rho \beta (T - T_l) g_i = 0 \ , \tag{3}$$

$$\frac{\partial T}{\partial t} + V_j T_{,j} - \zeta T_{,jj} = 0 \ , \tag{4}$$

where

$$\beta = -\frac{1}{\rho} \frac{\partial \rho}{\partial T} \bigg)_P \ , \tag{5}$$

and

$$\tau_{ij} = \mu (V_{i,j} + V_{j,i}) \ . \tag{6}$$

Note that the Boussinesq approximation is applied to restrict the thermal dependence of density to the body force term in equation (3). Although the scaling results in section V will show that buoyancy may be ignored for pore length scales $\leq 10^2$ μm, the approximation is useful for separating out hydrostatic pressure P_h from flow-induced pressure variations (i.e., dynamic pressure) P_d, where $P = P_h + P_d$.

In addition to the governing equations, there are several jump boundary conditions that apply along the meniscus surface. The simplest condition is the jump mass balance. It is used to relate interfacial mass flux to temperature and derive velocity conditions on boundaries 2 and 4. The balance[47] is expressed in terms of the scalar j by:

$$j = \rho \left(V_j - V_j^{(s)} \right) n_j = \rho^{(v)} \left(V_j^{(v)} - V_j^{(s)} \right) n_j \ , \tag{7}$$

$V_j^{(v)}$ is the vapor velocity, $V_j^{(s)}$ is the velocity of the interface, and the vapor density $\rho^{(v)}$ is treated as a constant. Assuming that the liquid domain Ω deforms with time, the total mass balance becomes:

$$\frac{d}{dt} \int_\Omega \rho \, d\Omega + \int_\Gamma j \, d\Gamma = 0 \ , \tag{8}$$

where Γ is the total boundary area. Applying incompressibility and restricting deformation and interfacial flux to boundary 4 yields the following equation for volumetric flowrate and velocity along surface 2:

$$-\int_{\Gamma_2} V_i n_i \, d\Gamma = \int_{\Gamma_4} \left(V_i - V_i^{(s)} \right) n_i \, d\Gamma + \frac{d\Omega}{dt} \ . \tag{9}$$

17

The second and probably most important jump condition is the momentum balance, which because of its directional dependence, consists of a normal and tangential component. Substitution of equation (7) into the normal component yields a general relationship for surface curvature which is ultimately used to calculate $y^{(s)}$. Similarly, from the tangential component one obtains an expression for thermocapillary stress and a condition for velocity along the meniscus. From reference 48, the complete tensoral expression for the jump momentum balance is:

$$j(V_i-V_i^{(v)}) - (\sigma_{ij}-\sigma_{ij}^{(v)}) \, n_j + \gamma \kappa n_i + \gamma_{,i} = 0 \ . \tag{10}$$

Equation (10) equates the stress imposed on the meniscus to interfacial momentum flux, pressure due to surface curvature κ, and tangential stress due to surface tension gradient. The normal and tangential components are obtained by projecting equation (10) onto the unit vectors n_i and s_i, respectively. In the tangential direction, one has:

$$j(V_i-V_i^{(v)}) \, s_i - (\sigma_{ij}-\sigma_{ij}^{(v)}) \, n_j s_i + \gamma_{,i} s_i = 0 \ . \tag{11}$$

Pressure cancels out of the stress tensor term because it acts normal to the surface. In addition, invoking continuity between the tangential components of liquid and vapor velocity, that is:

$$(V_i-V_i^{(v)}) \, s_i = 0 \ , \tag{12}$$

yields an equation for the tangential stress balance:

$$(\tau_{ij}-\tau_{ij}^{(v)}) \, n_j s_i = \gamma_{,i} s_i \ , \tag{13}$$

where

$$(\tau_{ij}-\tau_{ij}^{(v)}) \, n_j s_i = \mu \big[(V_{i,j}+V_{j,i}) - f_\mu (V_{i,j}^{(v)}+V_{j,i}^{(v)}) \big] n_j s_i \ ,$$

and f_μ is the ratio of vapor viscosity to liquid viscosity. An important aspect of the one-sided model is the assumption of a vanishingly small f_μ (reference 14) which removes vapor velocity from equation (13) and yields:

$$\tau_{ij} n_j = \gamma_{,i} \ . \tag{14}$$

The normal momentum/stress balance is obtained by projecting equation (10) onto the normal vector n_i.

$$j(V_i-V_i^{(v)}) \, n_i - (\sigma_{ij}-\sigma_{ij}^{(v)}) \, n_j n_i + \gamma \kappa = 0 \ . \tag{15}$$

The interfacial momentum flux is expressed as a function of j and the ratio of liquid density to vapor density $f_\rho \ (= \rho/\rho^{(v)})$ by:

$$j(V_i-V_i^{(v)}) \, n_i = \frac{j^2}{\rho} \, (1-f_\rho) \ . \tag{16}$$

In the case of evaporation, equation (16) represents the recoil force caused by vapor ejection from the surface. Because $f_\rho > 1$, the momentum of the vapor is higher than that of the liquid. Consequently, the momentum change and force exerted on the interface will always point into the

liquid. Equation (16) also represents the force resulting from condensation and vapor impingement on the surface. Because the vapor and liquid momentum vectors for this case point in a direction opposite of evaporation, the resultant force also points into the liquid. The independence of the recoil force to the direction of mass transfer is reflected by the squaring of j in equation (16). Actually, the direction only depends on the value of f_p, which for all liquid/vapor systems is greater than unity.

The stress tensor contribution in equation (15) is expressed in terms of pressure and viscous stress by:

$$(\sigma_{ij} - \sigma_{ij}^{(v)}) \, n_j n_i = - (P_d + P_h - P^{(v)}) + (\tau_{ij} - \tau_{ij}^{(v)}) \, n_j n_i \ . \tag{17}$$

One again refers to the definition of the difference between the liquid and vapor viscous stress tensors and applies the one-sided assumption, $f_\mu = 0$, to remove vapor velocity. This yields:

$$(\tau_{ij} - \tau_{ij}^{(v)}) \, n_j n_i = 2\mu V_{i,j} n_j n_i \ . \tag{18}$$

Substitution of equations (16) to (18) into the original equation for normal stress, equation (15), yields an expression for surface curvature that accounts for pressure, vapor recoil, and viscous stress:

$$\gamma \kappa = (P^{(v)} - P_d - P_h) + \frac{j^2}{\rho} \, (f_p - 1) + 2\mu V_{i,j} n_j n_i \ . \tag{19}$$

The third type of jump condition relates the transfer of latent heat and kinetic energy to the temperature gradient normal to the meniscus. In subsequent comparisons between thermocapillary and interfacial convection, it is necessary to impose equivalent heat transfer conditions regardless of the rate of mass transfer normal to the surface. That is, a method of accounting for convective energy transport in the case of pure thermocapillary flow is required. This is accomplished by expressing the energy jump in the general form:

$$q + (kT_{,i} n_i - k^{(v)} T_{,i}^{(v)} n_i) = 0 \ . \tag{20}$$

The one-sided approximation is applied by factoring out k and assuming that the ratio $k^{(v)}/k$ $\rightarrow 0$.[14] This yields the simplified relation:

$$q + kT_{,i} n_i = 0 \ . \tag{21}$$

q represents the local interfacial heat flux. One can define q in a general way by expressing it in terms of Newton's law of cooling, that is:

$$q = h(T - T_0) \ . \tag{22}$$

q can also be represented more specifically as the sum of latent heat and kinetic energy transported across the interface, that is:

$$q = j \left\{ L + \frac{1}{2} \left[(V_i^{(v)} - V_i^{(s)}) \, n_i \right]^2 - \frac{1}{2} \left[(V_i - V_i^{(s)}) \, n_i \right]^2 \right\} \ . \tag{23}$$

Note that viscous energy dissipation due to surface stress is ignored. The velocity differences in equation (23) can also be replaced by the definition for mass flux in equation (7). Substitution of these definitions for q into equation (21) yields two forms of the energy boundary condition, namely:

$$kT_{,i} + h(T - T_0) = 0 , \tag{24}$$

and

$$kT_{,i} + j \left\{ L + \frac{j^2}{2\rho^2} (f_\rho^2 - 1) \right\} = 0 . \tag{25}$$

Additional constitutive relations are required to account for the thermal dependency of surface tension and the sensitivity of mass flux to temperature along the surface. An equation of state for surface tension is obtained by treating γ as a monotonically decreasing function of temperature.

$$\gamma = \gamma_l - \left| \frac{\partial \gamma}{\partial T} \right| (T - T_l) . \tag{26}$$

γ_l is the surface tension at the minimum cavity temperature T_l. This formulation is typically used since $d\gamma/dT$ is negative for nearly all common liquids.

To relate mass flux to temperature, i.e., characterize the mechanism for evaporation and condensation, one begins with the well-known Hertz-Knudsen relation[49] which relates mass flux to the difference between the density of the vapor undergoing phase change $\rho^{(v)}$ and the density of the surrounding vapor $\rho_0^{(v)}$.

$$j = e \left\{ \frac{R_g T_0}{2\pi M_w} \right\}^{1/2} (\rho^{(v)} - \rho_0^{(v)}) . \tag{27}$$

$\rho^{(v)}$ is the vapor density at the liquid surface temperature T, while $\rho_0^{(v)}$ is the vapor density at T_0. e is the accommodation coefficient and is a factor representing the resistance to mass transfer. For a surface free from contaminants, it is assumed to be equal to 1. Other variables in equation (27) include the vapor molecular weight M_w and universal gas constant R_g. It is assumed that the saturation density and pressure are solely functions of temperature. Treating density as a linear function of temperature allows one to express the density difference as:

$$\rho^{(v)} - \rho_0^{(v)} = \frac{\partial \rho_0^{(v)}}{\partial T} (T - T_0) . \tag{28}$$

From the chain rule:

$$\frac{\partial \rho_0^{(v)}}{\partial T} = \frac{\partial \rho_0^{(v)}}{\partial P} \frac{dP}{dT} . \tag{29}$$

Assuming a perfect gas equation of state yields an expression for $\partial \rho_0^{(v)}/\partial P$:

$$\frac{\partial \rho_0^{(v)}}{\partial P} = \frac{M_w}{R_g T_0} , \tag{30}$$

20

dP/dT in equation (29) is simplified by applying the Clapyeron equation and assuming that $f_\rho \gg 1$. This yields:

$$\frac{dP}{dT} = \frac{L\rho_0^{(v)}}{T_0} \ .$$

(31)

Substitution of equations (28) to (31) into equation (27) yields a final relation for mass flux in terms of the temperature difference between the liquid surface and vapor.

$$j = \frac{e\rho_0^{(v)}L}{T_0^{3/2}}\left(\frac{M_w}{2\pi R_g}\right)^{1/2}(T-T_0) \ .$$

(32)

The remaining boundary conditions are Dirichlet-type and, apart from scaling, require no further simplification.

D. Scaling

One now nondimensionalizes and simplifies the governing equations and boundary conditions by applying different scaling factors. Length is scaled to the pore width D, and viscous scales are chosen for time, velocity, stress, pressure, and mass flux,[14] that is:

$$x_i = Dx_i^* \ , \qquad V_i = \frac{v}{D}V_i^* \ , \qquad t = \frac{D^2}{v}t^* \ ,$$

$$g_i = a_i g_0 \ , \qquad \sigma_{ij} = \frac{\rho v^2}{D^2}\sigma_{ij}^* \ , \qquad P = \frac{\rho v^2}{D^2}P^* \ ,$$

(33)

$$j = \frac{\rho v}{D}j^* \ .$$

These scales are appropriate for a system in which viscosity and surface tension influence the flowfield. They should also apply in instances where the domain is nonisothermal and interfacial mass transfer is not too intense. For temperature, one scales the difference $T-T_0$ to the maximum difference in the cavity $|\Delta T| = |T_1-T_0|$. This yields the relationship:

$$T = T^*|\Delta T|+T_0 \ .$$

(34)

The appeal in using this definition is that the scaled vapor temperature assumes the same value of 0 for both superheating and subcooling. With superheating the side wall temperature is fixed at $T^* = 1$, while with subcooling, it is held at $T^* = -1$.

Applying the scaling factors in equations (33) and (34) yields the following dimensionless forms of the governing equations:

$$V_{i,i}^* = 0 \ , \tag{35}$$

$$\frac{\partial V_i^*}{\partial t^*} + V_j^* V_{i,j}^* + P_{d,i}^* - \tau_{ij,j}^* - Gra_i T^* = 0 \ , \tag{36}$$

$$\frac{\partial T^*}{\partial t} + V_j^* T_{,j}^* - \frac{1}{Pr} T_{,jj}^* = 0 \ . \tag{37}$$

The two dimensionless parameters arising from the scaling are:

$$Gr = \frac{D^3 g \beta |\Delta T|}{v^2} \ , \qquad \text{Grashoff number} \tag{38}$$

$$Pr = \frac{v}{\varsigma} \ . \qquad \text{Prandtl number} \tag{39}$$

Scaling of the constitutive relations for mass flux and surface tension are critical to formulating a working set of equations. The constitutive equation for mass flux is scaled to temperature by substituting equation (34) into equation (32) to yield:

$$j^* = \frac{T^*}{Rs} \ , \tag{40}$$

where

$$Rs = \frac{f_\rho v T_0^{3/2}}{eDL|\Delta T|} \left\{ \frac{2\pi R_g}{M_W} \right\}^{1/2} \ . \tag{41}$$

The interfacial resistance Rs defines the degree of nonequilibrium (i.e., difference in state between the liquid and vapor) that can be maintained per unit mass undergoing phase change at a volatile interface. $Rs = 0$ corresponds to the quasi-equilibrium limit, where the interfacial temperature is constant and equal to the saturation value, $T^* = 0$. $1/Rs = 0$ corresponds to the nonvolatile case in which the evaporation mass flux j^* is zero.

A dimensionless equation of state for surface tension is obtained by substituting equation (34) into equation (26). This yields two equations which may be expressed in the single general form as:

$$\gamma = \gamma_l (1 - Cr T^*) \tag{42}$$

where

$$Cr = \frac{\left| \frac{\partial \gamma}{\partial T} \right| |\Delta T|}{\gamma_l} \ . \tag{43}$$

T^* is a corrected scaled temperature whose value depends on the maximum and minimum temperature in the cavity. For superheating ($T_h = T_1$ and $T_1 = T_0$), $T^* = T^*$, while for subcooling ($T_h = T_0$ and $T_1 = T_1$), $T^* = T^* + 1$. The Crispation number Cr denotes the sensitivity of surface tension to temperature. Because it is quite low for most ordinary fluids, several studies have justified neglect of surface deformation by assuming a very small Cr. Although several researchers, such as Sen and Davis[24] and Zebib et al.,[27] refer to Cr as the Capillary number, the terminology of Strani et al.[26] will be followed.

22

In scaling the normal stress balance, one sets the hydrostatic pressure at the base of the cavity to a reference value of 0. The hydrostatic pressure along the meniscus P_h can then be expressed as a function of $y^{(s)}$:

$$P_h = -\rho g y^{(s)} .$$

(44)

P_h is negative because the body force acceleration points in the $-x_2$-direction. Substituting equations (40), (42), and (44) into equation (19), and rearranging terms yields the dimensionless equation for surface curvature:

$$\kappa^* = \frac{B_o y^{(s)*} - Ca(P_d^* - P^{(v)*}) + VrT^{*2} + 2CaV_{i,j}^* n_j n_i}{1 - CrT^*} .$$

(45)

Previously undefined parameters arising from this scaling are:

$$Bo = \frac{\rho g D^2}{\gamma_l} ,$$ static Bond number   (46)

$$Ca = \frac{\rho v^2}{\gamma_l D} ,$$ Capillary number   (47)

$$Vr = \frac{Ca(f_\rho - 1)}{Rs^2} .$$ recoil parameter   (48)

Ca denotes the force ratio between viscosity and surface tension. Although it appears to have a significant effect on surface geometry, Ca, like Cr, is typically small, particularly for high surface tension liquids. Note that this dimensionless grouping has previously been referred to as the Ohnesorge number Oh by Cuvilear and Driessen[29] and the nondimensional surface tension number by other researchers.

For the shear-stress condition, one expands the surface tension gradient by applying the chain rule:

$$\gamma_{,i} = \frac{\partial \gamma}{\partial T} \frac{\partial T}{\partial x_i} .$$

(49)

Substituting equation (49) into equation (14) and scaling yields:

$$\tau_{ij}^* n_j = -ReT_{,i}^* .$$

(50)

Re is the surface tension Reynolds number and is defined as:

$$Re = \frac{\left|\frac{\partial \gamma}{\partial T}\right| D |\Delta T|}{\rho v^2} .$$

(51)

Since the normal gradients of V_1^* and V_2^* are interdependent, the condition represented by equation (50) can only be applied in one direction. That is, the stress conditions in both directions are mutually satisfied by applying the Neumann condition to either V_1^* or V_2^*. If equation (50) is

applied to V_1*, a Dirichlet condition for V_2* is obtained by substituting equation (7) into equation (32) and scaling terms:

$$V_2^* = \frac{1}{n_2}\left(\frac{T^*}{Rs} - V_1^* n_1\right) .$$ (52)

The equation used to specify velocity along boundary 2, equation (9), is also scaled to yield:

$$-\int_{\Gamma_2^*} V_i^* n_i d\Gamma^* = \int_{\Gamma_4^*} (V_i^* - V_i^{(s)*}) n_i d\Gamma^* + \frac{d\Omega^*}{dt^*} .$$ (53)

The result of scaling the two jump energy equations, equations (24) and (25), can be expressed as the general relationship:

$$T_{,i}^* n_i = -BiT^* .$$ (54)

Bi is the effective Biot number of the liquid volume. For the case in which heat transfer is expressed in terms of Newton's law of cooling, Bi is defined as the convective Biot number, namely:

$$Bi = Bi_c = \frac{hD}{k} .$$ (55)

For the case involving latent heat and kinetic energy transport, Bi consists of the sum:

$$Bi = Bi_m + Bi_{ke} T^{*2} ,$$ (56)

where

$$Bi_m = \frac{1}{RsE} ,$$ (57)

$$Bi_{ke} = \frac{(f_\rho^2 - 1)}{2\Lambda Rs^3 E} ,$$ (58)

$$E = \frac{k|\Delta T|}{\rho vL} ,$$ (59)

and

$$\Lambda = \frac{D^2 L}{v^2} .$$ (60)

The rate of latent heat transfer relative to heat conduction is represented by the mass transfer Biot number Bi_m. In a manner similar to Bi_c, the value of Bi_m indicates whether heat transfer is limited by conduction within the liquid ($Bi_m > 1$) or mass transfer at the interface ($Bi_m < 1$).

The influence of mass transfer and phase change on the interfacial temperature distribution can also be viewed in terms of the reciprocal product, RsE. The parameter E represents the ratio between the viscous and evaporation/condensation time scales (i.e., D^2/v and $\rho D^2 L/(k|\Delta T|)$, respectively). Although it applies to both modes of interfacial transfer, E has been termed the evaporation number by Burelbach et al.,[14] and it is indicative of the rate of phase change relative to momentum diffusion in the cavity.

A large value of E signifies a high rate of evaporation and shorter timescale relative to viscous effects. This is comparable to having a high ratio between the thermal conductivity and

24

convective heat transfer coefficient for Bi_c. In either case, raising k relative to latent heat transport in equation (57) or h in equation (55) means that interfacial convection progressively becomes the rate limiting process for heat transfer.

Interfacial resistance can either augment or restrict the effect of phase change on rate limiting heat transfer processes. Rs dictates the rate at which vaporized liquid or condensed vapor is ejected from or deposited onto the interface. Hence a higher rate, reflected by lower Rs, would suppress the influence of phase change on the temperature profile. That is, the rate limiting effect of phase change relative to conduction is reduced by increasing the mass flowrate into or out of the cavity. This is reflected by a higher value of Biot number Bi_m.

The kinetic energy Biot number $Bi_{ke}T^{*2}$ represents the rate of kinetic energy transfer relative to conduction at the surface. Its contribution to the energy jump balance is evaluated by factoring out Bi_e from equation (56), that is:

$$Bi = Bi_e\left(1 + \frac{Bi_{ke}}{Bi_e} T^{*2}\right) . \tag{61}$$

Note that Λ, which arises in the denominator of $Bi_e/Bi_{ke} = (f_\rho^2-1)/(2\Lambda R_s^2)$, is extremely large for most fluids. Consequently, we can safely assume that $Bi_e/Bi_{ke} \approx 0$. This conveniently enables one to use the same linear relationship, equation (54), for both forms of the boundary 4 energy equation. One now has two definitions for the Biot number at the surface. Both of these are equivalent in terms of dictating temperature distribution along the interface.

Scaling the governing equations and boundary conditions has yielded 13 dimensionless groupings based on 14 thermophysical properties and a contact angle constraint on either side wall. Two of the groupings, Vr and Bi_{ke}, are immediately recognized as combinations of the other 11. Five of these remaining 11 (i.e., Gr, Re, Ca, Bo, and Cr) represent ratios between the five principal modes of force application, namely viscous stress, surface tension, gravity, thermocapillarity, and buoyancy. Scaling considerations show that only four of these ratios are mutually independent and compatible with a consistently defined problem. Note that $Ca = Cr/Re$, which reduces the number of dimensionless groupings to 10.

All of the remaining dimensionless force groups (i.e., Gr, Re, Bo, and Cr) are dependent on cavity width. Although the relationship between thermocapillarity and buoyancy can be indirectly expressed by the relative magnitudes of Re and Gr, no parameter has been defined that directly relates these effects on purely thermophysical terms, i.e., as a function of their relative sensitivities to temperature. This is possible because both are expressed as gradients with respect to temperature. The so-called thermal response ratio is defined:

$$\lambda = \frac{\gamma_t \beta}{|\partial \gamma/\partial T|} , \tag{62}$$

which relates the thermal sensitivity of surface tension to that of density. It is appealing to use λ, because, like Pr, it is a physical property of the fluid and independent of imposed temperatures and scaling dimensions. For most Newtonian fluids, it ranges between 0.1 to 0.01, and can be related to the other dimensionless parameters by:

$$Gr = BoRe\lambda . \tag{63}$$

It was decided to replace Gr with equation (63) in the momentum equation because in most cases thermocapillarity predominates fluid motion. Note that the group $\lambda Bo = Bo_d$ is sometimes referred to as the dynamic Bond number.[21] Also, to eliminate confusion between Re and the well-known parameter relating viscous and inertial effects, Re is replaced with the Marangoni number Ma, defined as:

$$Ma = RePr .$$ (64)

E. Summary

The theoretical development has produced a mathematical model consisting of the governing equations and boundary conditions summarized in table 1. Of the five independent variables in this system of equations, only pressure lacks any specific boundary conditions. Section IV will discuss how the selected solution approach obviates the need for a pressure boundary condition at more than two points.

Because the cavity is symmetric, only two conditions are needed to define surface geometry. One is the contact angle constraint and the other is the requirement that the distance from the contact line to the cavity base be equivalent to the width.

In the process of scaling these equations, we also obtained a variety of dimensionless groupings. The ones derived in this chapter and those referred to in other sections of the text are presented in table 2.

IV. NUMERICAL METHOD

A. Introduction

Except for the complications introduced by the existence of an unknown free boundary, the aforementioned problem can be readily solved using various numerical techniques. The finite element method has been selected because of its strength in handling unstructured grid domains and free boundaries. Although it is possible to structure a numerical solution so that the surface is determined simultaneously with V_i, P_d, and T, this approach is memory-intensive and requires manipulation of an extremely large sparse coefficient matrix. The preferred approach[29][35] relaxes the normal-stress boundary condition in equation (45) and calculates surface position $y^{(s)}$ in an iteration loop outside the steady-state flowfield solution.

Note that the * notation which signified scaled parameters in section III has been dropped. Unless specified otherwise, all variable and parameters references in the remaining sections pertain to their dimensionless form.

B. Finite Element Equations

Finite element equations for the flow variables are obtained by applying the Galerkin method to derive weak forms of the scaled momentum and energy equations, equations (34) and (35), respectively. These equations are discretized with respect to time via a semi-implicit Crank-Nicolson scheme, while continuity, equation (35), is enforced using a SIMPLER-type algorithm similar to the finite volume-based technique of Patankar et al.[50]

Table 1. Summary of governing equations and boundary conditions.

Domain Equations	Description
$V_{i,i} = 0$	Continuity equation (equation (35)). Used with momentum to solve velocity and pressure.
$\dfrac{\partial v_i}{\partial t^*} + V_j V_{i,j} + P_{d,i} - \tau_{ij,j} - Gra_i T = 0$	Momentum equation (equation (36)). Used with continuity to solve velocity and pressure.
$\dfrac{\partial T}{\partial t} + V_j T_{,j} - \dfrac{1}{Pr} T_{,jj} = 0$	Energy equation (equation (37)). Used to solve temperature.
$\kappa = \dfrac{1}{1-CrT}\left[\begin{array}{l}Boy^{(s)} - Ca(P_d - P^{(v)}) \\ +VrT^2 + 2CaV_{i,j}n_j n_i\end{array}\right]$	Equation for meniscus curvature (equation (45)). Used to solve $y^{(s)}$ when κ replaced by expression for curvature in 2-D Cartesian frame.

Boundary 1 Conditions	Description
$V_1 = 0$	Impermeability
$V_2 = 0$	No-slip
$T = T_1$	Isothermal side wall. $T_1 = 1$ (superheating), $T_1 = -1$ (subcooling).

Boundary 2 Conditions	Description
$V_1 = 0$	Parallel flow
$V_2 = \dfrac{1}{\Gamma_2}\displaystyle\int_{\Gamma_4}\dfrac{T}{Rs}\,d\Gamma_4$	Uniform flow (equation (53)). Total flowrate matches value across meniscus.
$T = T_1$	Isothermal. Same temperature as side walls.

Boundary 3 Conditions	Description
$V_1 = 0$	Impermeability
$V_2 = 0$	No-slip
$T = T_1$	Isothermal side wall. $T_1 = 1$ (superheating), $T_1 = -1$ (subcooling).

Boundary 4 Conditions	Description
$\tau_{ij}n_j = -ReT_{,i}$	Thermocapillary stress condition (equation (50)).
$V_2 = \dfrac{1}{n_2}\left(\dfrac{T}{Rs} - V_1 n_1\right)$	Required V_2 needed to satisfy meniscus flux condition (equation (52)).
$T_{,i}n_i = -BiT$	Robin energy condition (equation (54))
$y^{(s)} = 1$	Surface Dirichlet condition @ $x_1 = 0$ and 1
$\left|\dfrac{dy^{(s)}}{dx_1}\right| = \tan(\pi/2 - \omega)$	Surface Neumann condition @ $x_1 = 0$ and 1

Note: * notation dropped. All variables expressed in dimensionless form.

Table 2. Summary of dimensionless groupings.

Group	Name	Description	Formula
Bi	Biot number	General reference for either Bi_c or Bi_m (equation (54))	Either Bi_c or Bi_m
Bi_c	Convective Biot number	Ratio of effective surface heat transfer convection to conduction (equation (55))	$= \dfrac{hD}{k}$
Bi_m	Mass Transfer Biot number	Ratio of latent heat transport to conduction (equation (57))	$= \dfrac{1}{RsE}$
Bi_{ke}	Kinetic energy Biot number	Ratio of kinetic energy transport to conduction (equation (58))	$= \dfrac{(f_\rho^2 - 1)}{2\Lambda Rs^3 E}$
Bo	Bond number	Ratio of gravitational to surface tension forces (equation (46))	$= \dfrac{\rho g D^2}{\gamma}$
Bo_d	Dynamic Bond number	Ratio of buoyancy to thermocapillary forces (equation (63))	$= \dfrac{gD^2 \partial\rho/\partial T}{\|\partial\gamma/\partial T\|}$ $= Bo\lambda$
Ca	Capillary number	Ratio of viscous to surface tension forces (equation (47))	$= \dfrac{\rho v^2}{\gamma D}$ $= Cr/Re$
Cr	Crispation number	Ratio of thermocapillary to surface tension forces (equation (43))	$= \dfrac{1}{\gamma}\left|\dfrac{\partial\gamma}{\partial T}\right|\|\Delta T\|$ $= CaRe$
E	Evaporation number	Ratio between viscous and evaporation/ condensation time scales (equation (59))	$= \dfrac{k\|\Delta T\|}{\rho vL}$
f_ρ	Density ratio	Ratio of liquid to vapor density (equation (16))	$= \rho/\rho^{(v)}$
Gr	Grashoff number	Ratio of buoyancy to viscous forces (equation (38))	$= \dfrac{D^3 g\beta \|\Delta T\|}{v^2}$ $= Bo\lambda Re$
λ	Thermal response ratio	Ratio of density to surface tension thermal sensitivities (equation (62))	$= -\dfrac{\gamma}{\rho}\dfrac{\partial\rho/\partial T}{\|\partial\gamma/\partial T\|}$
Λ	Dimensionless latent heat	Latent heat scaled according to viscous scales (equation (60))	$= \dfrac{D^2 L}{v^2}$
Ma	Marangoni number	Ratio of thermocapillary stress force to viscous forces (equation (64))	$= \dfrac{\left|\dfrac{\partial\gamma}{\partial T}\right| D\|\Delta T\|}{\mu\zeta}$ $= RePr$

Table 2. Summary of dimensionless groupings (continued).

Group	Name	Description	Formula
Oh	Ohnesorge number	Ratio of viscous to surface tension forces[29]	$= \left(\dfrac{\mu v}{\gamma D}\right)^{1/2}$ $= \sqrt{Ca}$
Pr	Prandtl number	Ratio of momentum to thermal diffusivity (equation (39))	$= v/\zeta$
Re	Surface tension Reynolds number	Ratio of thermocapillary to viscous forces (equation (51))	$= \dfrac{\left\lvert\dfrac{\partial\gamma}{\partial T}\right\rvert D\lvert\Delta T\rvert}{\rho v^2}$ $= Ma/Pr$
Rs	Interfacial resistance	Degree of nonequilibrium between vapor and liquid on free surface (equation (41))	$= \dfrac{f_p v T_0^{3/2}}{eD U\lvert\Delta T\rvert}\left(\dfrac{2\pi R_g}{M_w}\right)^{1/2}$
Sv	Viscous stress parameter	Ratio of normal velocity gradient to temperature at surface (equation (8))	$= Br/Rs$
Vr	Recoil parameter	Ratio of surface momentum change to surface tension (equation (48))	$= \dfrac{Ca(f_p-1)}{Rs^2}$

SIMPLER requires three additional Galerkin equations for the pressure and velocity corrections P_d' and V_i', respectively. These equations are derived from equations (35) and (36), which for two-dimensions, represent a system consisting of three equations and three unknowns. Because equation (36) is actually two equations, it is possible to derive velocity estimates V_i^o with an arbitrary value of pressure P_d^o which may not satisfy continuity. This implies the existence of corrections which, when added to the estimated values, mutually satisfy equations (35) and (36), where $V_i = V_i' + V_i^o$ and $P_d = P_d' + P_d^o$. Substituting these expressions into equation (36) and subtracting the momentum equation based on estimated values of pressure and velocity yields an expression for V_i' in terms of V_i^o, P_d^o, and P_d':

$$\frac{\partial V_i'}{\partial t} + V_j' V_{i,j}^o + V_j' V_{i,j}' + V_j^o V_{i,j}' + P_{d,i}' - V_{i,jj}' = 0 \ . \tag{65}$$

By approximating $\partial V_i'/\partial t$ as a finite difference and assuming that V_i' at the previous time step n equals zero, one finds that $\partial V_i'/\partial t = V_i'/\Delta t$. Substituting this approximation into equation (65) yields a quasi-explicit expression for V_i'. As an initial estimate, all second-order and nonlinear terms are neglected, yielding:

$$V_i' = -\Delta t P_{d,i}' \ . \tag{66}$$

The continuity constraint is applied by substituting the sum of equation (66) and V_i^o into equation (35), and rearranging terms to yield an expression for the pressure correction that enforces continuity:

$$P_{d,ii}' = \frac{V_{i,i}^o}{\Delta t} \ . \tag{67}$$

With SIMPLER, a more accurate P_d' is obtained prior to the pressure update. This improved estimate is derived by invoking continuity with the complete equation for V_i'. Substituting $V_i = V_i' + V_i^o$ into the continuity equation as before yields the more accurate expression:

$$P_{d,ii}' = \frac{V_{i,i}^o}{\Delta t} - 2V_{i,j}'V_{j,i}^o - V_j^o V_{i,ji}' - V_{i,j}'V_{j,i}' + V_{i,jji}' .$$

(68)

The pressure correction calculated from equation (68) is added to the previous estimate P_d^o and used in the next calculation of V_i^o. Use of SIMPLER increases the total number of equations and unknowns from four to six. Note that P_d is no longer a dependent variable in the true sense since it represents a cumulative sum of P_d' estimates. Also, equations (67) and (68) count as only one equation.

Algebraic finite element equations for these six variables are obtained by applying the Galerkin method.[51] First, the momentum, energy, and correction equations are expressed in residual form:

$$\varepsilon_i^M = \frac{\partial V_i}{\partial t} + V_j V_{i,j} + P_{d,i} - \tau_{ij,j} - Gra_i T , \qquad \text{momentum} \qquad (69)$$

$$\varepsilon^E = \frac{\partial T}{\partial t} + V_j T_{,j} - \frac{1}{Pr} T_{,jj} , \qquad \text{energy} \qquad (70)$$

$$\varepsilon^{\Delta P1} = P_{d,ii} - \frac{V_{i,i}}{\Delta t} , \qquad P_d \text{ correction (initial)} \qquad (71)$$

$$\varepsilon_i^{\Delta V} = V_i' + \Delta t P_{d,i}' , \qquad V_i \text{ correction} \qquad (72)$$

$$\varepsilon^{\Delta P2} = P_{d,ii}' - \frac{V_{i,i}}{\Delta t} + 2V_{i,j}'V_{j,i} + V_j V_{i,ji}' + V_{i,j}'V_{j,i}' - V_{i,jji}' . \qquad P_d \text{ correction (final)} \qquad (73)$$

Note, the o-superscript has been eliminated by consolidating the definition of estimated velocity and pressure, that is $V_i = V_i^o$ and $P_d = P_d^o$. Weak forms of the governing differential equations E_α are derived by integrating the residuals ε over the spatial Ω and temporal φ domains:

$$E_\alpha = \int_0^1 U(\varphi) \int_\Omega W_\alpha \varepsilon \, d\Omega d\varphi = \int_0^1 U(\varphi) S_\alpha d\varphi = 0 . \qquad (74)$$

Prior to each integration, the residual is multiplied by test functions that characterize the dependent variable's spatial variance between adjacent nodes W_α and its change between successive temporal stations $U(\varphi)$. Integration of the residuals in equations (69) to (73) over Ω yields the corresponding equations:

$$S_{\alpha i}^M = \int_\Omega W_\alpha^M \left(\frac{\partial V_i}{\partial t} + V_j V_{i,j} + P_{d,i} - GrTa_i \right) d\Omega + \int_\Omega W_{\alpha,j}^M (V_{i,j} V_{j,i}) d\Omega - \int_\Gamma \dot{W}_\alpha^M \tau_{ij} n_j d\Gamma , \qquad (75)$$

$$S_\alpha^E = \int_\Omega W_\alpha^E \left(\frac{\partial T}{\partial t} + V_j T_{,j}\right) d\Omega + \frac{1}{Pr} \int_\Omega W_{\alpha,j}^E T_{,j} d\Omega - \frac{1}{Pr} \int_\Gamma \dot{W}_\alpha^E T_{,j} n_j d\Gamma , \tag{76}$$

$$S_\alpha^{\Delta P1} = \int_\Omega W_{\alpha,i}^{\Delta P1} P_{d,i}' d\Omega - \int_\Gamma \dot{W}_\alpha^{\Delta P1} P_{d,i}' n_i d\Gamma + \frac{1}{\Delta t} \int_\Omega W_\alpha^{\Delta P1} V_{i,i} d\Omega , \tag{77}$$

$$S_{\alpha i}^{\Delta V} = \int_\Omega W_\alpha^{\Delta V} (V_i' + \Delta t P_{d,i}') d\Omega , \tag{78}$$

$$S_\alpha^{\Delta P2} = \int_\Omega W_{\alpha,i}^{\Delta P2} P_{d,i}' d\Omega - \int_\Gamma \dot{W}_\alpha^{\Delta P2} P_{d,i}' n_i d\Gamma + \int_\Omega W_\alpha^{\Delta P2} \left(\frac{V_{i,i}}{\Delta t} - 2V_{i,j}' V_{j,i} - V_j V_{i,ji}' - V_{i,j}' V_{j,i}' + V_{i,jji}'\right) d\Omega . \tag{79}$$

The viscous stress gradient in equation (69) is integrated by parts to yield a surface integral in equation (75) that represents imposed stress along the meniscus. Applying the same approach to equation (70), $T_{,jj}$ is integrated by parts to obtain a surface integral in equation (76) that specifies the temperature gradient normal to the boundary. In both pressure correction residuals, equations (77) and (79), the equation order is reduced by integrating $P_{d,ii}'$ by parts, and a surface integral is obtained by containing the normal gradient of P_d'. Since $P_d' = 0$ throughout Ω to satisfy both continuity and momentum, $P_{d,i}'$ must vanish along all boundaries. Note that no Neumann tensor arises in equation (78) from integration of equation (72).

Each flow variable and its associated gradient in equations (75) to (79) are discretized as a product of the nodal value and an interpolation function that characterizes its variance over the element domain.

$$V_i = \Phi_\beta V_{\beta i} , \quad V_{i,j} = \Phi_{\beta,j} V_{\beta i} ,$$

$$P = \Pi_\beta P_\beta , \quad P_{,i} = \Pi_{\beta,i} P_\beta , \tag{80}$$

$$T = \theta_\beta T_\beta , \quad T_{,i} = \theta_{\beta,i} T_\beta .$$

By decoupling meniscus geometry from the transient solution of V_i, P_d, and T, grid geometry can be treated as static with respect to the flowfield, and the interpolation functions depend solely on x_i. Note that a coupled and transient determination of meniscus geometry would require an accounting of grid convection due to interfacial deformation.

An example of a typical finite element domain consisting of 25 by 20 = 500 quadrilateral elements is shown in figure 5.

The different nodal configurations for these elements are illustrated in figure 6. The global coordinates x_i are assumed to vary quadratically with respect to the natural coordinate frame ξ_1 (ξ) and ξ_2 (η) centered at each element. For incompressible problems of this type, the traditional approach is to employ interpolation functions for velocity (Φ_β) and temperature (θ_β) that are one order higher than pressure (Π_β).
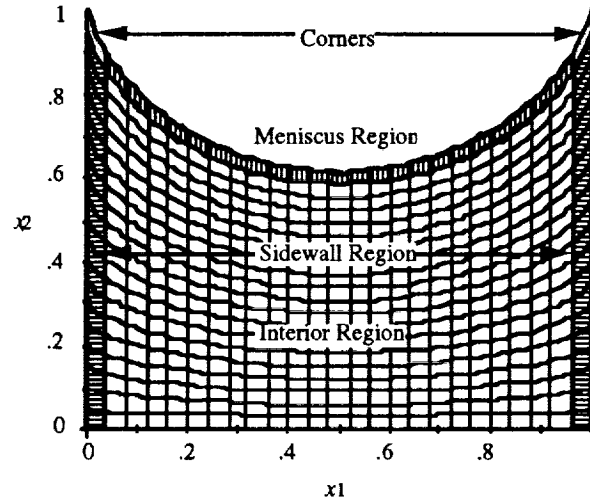
Figure 5. Finite element domain.
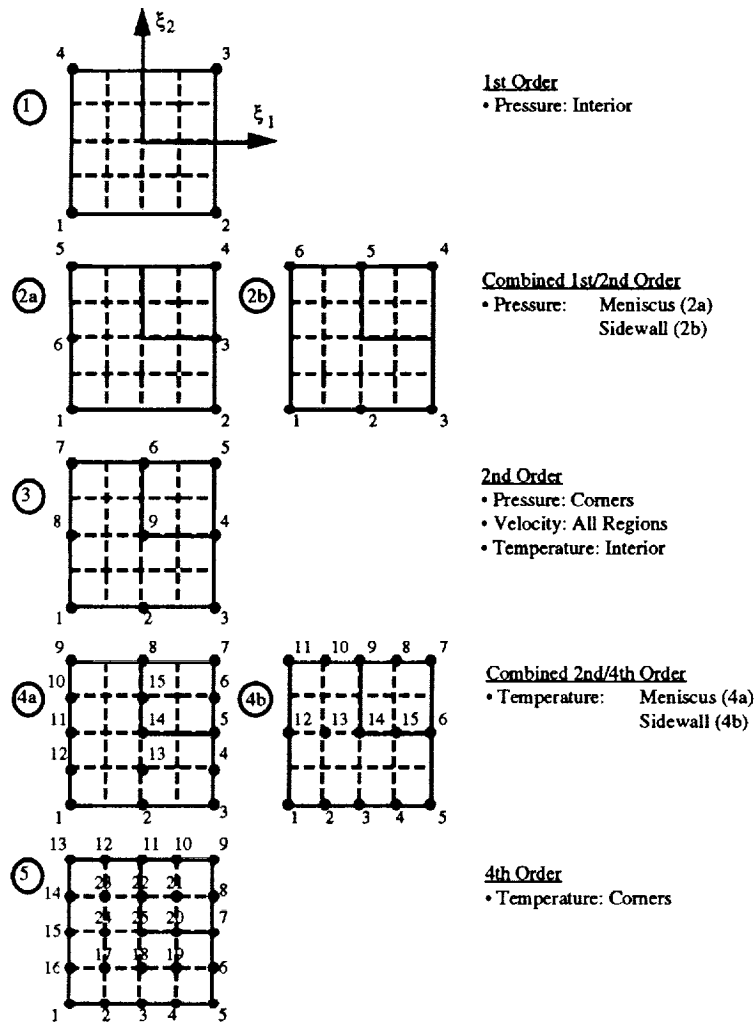
Figure 6. Element types and node definitions.

32

To accommodate the $V_i V_i{'}_{ji}$ term in the SIMPLER pressure update, equation (79), one follows the approach of other researchers and employs a quadratic variance for Φ_β and θ_β, and a linear variation for Π_β. This worked well in all regions except the meniscus interline. Here, thermocapillary stress and interfacial mass transport are greatest and cause substantial pressure gradients. Unless the corner elements are made sufficiently small, this abrupt change in pressure is inaccurately applied across the entire element and over-influences the velocities at adjoining nodes. A similar problem with temperature was encountered when analyzing high Bi ($\geq 10^2$). In this case, the second-order function is incapable of handling steep temperature gradients near the wall.

To correct this problem, more complex formulations for the pressure and temperature interpolations are used. Computer storage requirements are reduced and physical consistency is maintained between P_d and V_i by employing higher order interpolations in the corner and mixed order functions in the side wall and meniscus regions. This enables application of the standard first- and second-order formulations in the interior, but requires the seven element configurations shown in figure 6. A summary of the derivation of interpolation functions for these elements is given in appendix A.

For pressure, a quadratic approximation (i.e., type 3 in figure 6) is applied at the two corner elements, which have a nine-node element domain and coincide with the velocity interpolation. In the interior, however, a first-order variation (type 1) is retained to preserve the physical relationship between pressure and velocity. This approach requires the use of mixed order elements (types 2a and 2b) along the meniscus and side walls to ensure consistency between the number of local node contributions and global node equations. As detailed in appendix A, the type 2a function is applied along the meniscus, and is formulated at each of the six local nodes from the first and second-order Lagrange polynomials in the ξ_1 and ξ_2 directions, respectively. Alternatively, the type 2b interpolation is applied along the side walls, and is the product of the second and first-order Lagrange polynomials in the ξ_1 and ξ_2 directions, respectively.

A similar approach is taken with temperature. Originally, third-order and mixed third/second-order functions were investigated, but difficulties with convergence along the meniscus were encountered. Use of fourth and combined fourth/second-order interpolations, however, proved successful, and closely followed the methodology outlined for pressure.

The W_α test functions in equations (75) to (79) assume the same order as the interpolation associated with each equation's dependent parameter. Therefore, $W_\alpha^M = W_\alpha^{\Delta V} = \Phi_\alpha$, $W_\alpha^E = \theta_\alpha$, and $W_\alpha^{\Delta P1} = W_\alpha^{\Delta P2} = \Pi_\alpha$. Substitution of these definitions into equations (75) to (79) yields general expressions for the spatial residuals:

$$S_{\alpha i}^M = M_{\alpha\beta}^1 \frac{\partial V_{\beta i}}{\partial t} + (M_{\alpha\gamma\beta j}^2 V_{\gamma j} + M_{\alpha\beta}^3) V_{\beta i} + M_{\alpha j\beta i}^4 V_{\beta j} + M_{\alpha\beta i}^5 P_\beta - Gra_i M_{\alpha\beta}^6 T_\beta G_{\alpha i}^M \ , \tag{81}$$

$$S_\alpha^E = E_{\alpha\beta}^1 \frac{\partial T_\beta}{\partial t} + \left(E_{\alpha\gamma\beta j}^2 V_{\gamma j} + \frac{1}{Pr} E_{\alpha\beta}^3 \right) T_\beta - G_\alpha^E \ , \tag{82}$$

$$S_\alpha^{\Delta P1} = P_{\alpha\beta}^1 P_\beta' + \frac{1}{\Delta t} P_{\alpha\beta i}^2 V_{\beta i} \ , \tag{83}$$

$$S_{\alpha i}^{\Delta V} = M_{\alpha\beta}^1 V_{\beta i}' + \Delta t \, M_{\alpha\beta i}^5 P_\beta' \ , \tag{84}$$

33

$$S_\alpha^{\Delta P2} = P_{\alpha\beta}^1 P_\beta' + \frac{1}{\Delta t} P_{\alpha\beta i}^2 V_{\beta i} - P_{\alpha\beta j\gamma i}^3 V_{\beta i}'(2V_{\gamma j}+V_{\gamma j}') - P_{\alpha\beta\gamma i j}^4 V_{\beta j} V_{\gamma i}' , \qquad (85)$$

where

$$E_{\alpha\beta}^1 = \int_\Omega \theta_{,\alpha}\theta_{,\beta}d\Omega , \qquad\qquad M_{\alpha\beta}^1 = \int_\Omega \Phi_\alpha\Phi_\beta d\Omega , \qquad (86)$$

$$E_{\alpha\gamma\beta j}^2 = \int_\Omega \theta_{,\alpha}\Phi_\gamma\theta_{,\beta,j}d\Omega , \qquad\qquad M_{\alpha\gamma\beta j}^2 = \int_\Omega \Phi_\alpha\Phi_\gamma\Phi_{\beta,j}d\Omega ,$$

$$E_{\alpha\beta}^3 = \int_\Omega \theta_{,\alpha,j}\theta_{,\beta,j}d\Omega , \qquad\qquad M_{\alpha\beta}^3 = \int_\Omega \Phi_{\alpha,j}\Phi_{\beta,j}d\Omega ,$$

$$G_\alpha^E = \frac{1}{Pr}\int_\Gamma \dot\theta_{,\alpha}T_{,j}n_j d\Gamma , \qquad\qquad M_{\alpha j\beta i}^4 = \int_\Omega \Phi_{\alpha,j}\Phi_{\beta,i}d\Omega ,$$

$$P_{\alpha\beta}^1 = \int_\Omega \Pi_{\alpha,i}\Pi_{\beta,i}d\Omega , \qquad\qquad M_{\alpha\beta i}^5 = \int_\Omega \Phi_\alpha\Pi_{\beta,i}d\Omega ,$$

$$P_{\alpha\beta i}^2 = \int_\Omega \Pi_\alpha\Phi_{\beta,i}d\Omega , \qquad\qquad M_{\alpha\beta}^6 = \int_\Omega \Phi_\alpha\theta_{,\beta}d\Omega ,$$

$$P_{\alpha\beta j\gamma i}^3 = \int_\Omega \Pi_\alpha\Phi_{\beta,j}\Phi_{\gamma,j}d\Omega , \qquad\qquad G_{\alpha i}^M = \int_\Gamma \dot\Phi_\alpha\tau_{ij}n_j d\Gamma .$$

$$P_{\alpha\beta\gamma i j}^4 = \int_\Omega \Pi_\alpha\Phi_\beta\Phi_{\gamma,ji}d\Omega ,$$

The integral associated with $V_{i,jji}'$ in equation (79) is ignored, since it is impossible to discretize a third-order derivative using a second-order interpolation function. However, neglecting this term will only reduce the rate of convergence and should not alter the final solution since $V_i' \rightarrow 0$ near the proper value of pressure. All tensor values, except G_{ai}^M and G_a^E, are calculated using Gaussian quadrature in two dimensions. Because of the fourth-order interpolation in the E-tensor forms, a five-point procedure is employed.

The momentum and energy spatial residuals equations (81) and (82) are discretized with respect to time by integrating the product of $U(\varphi)$ and S_α in equation (74) over $0 \leq \varphi \leq 1$. φ is the normalized time increment and is defined as $\varphi = t/\Delta t$. Each nodal value of velocity $V_{\beta i}$ and temperature T_β (which are both generally expressed as Z_β) is discretized between the temporal stations, n and $n+1$ by $Z_\beta = \vartheta_n Z_\beta^n + \vartheta_{n+1} Z_\beta^{n+1}$, where $\vartheta_n = 1-\varphi$ and $\vartheta_{n+1} = \varphi$. This definition yields the dependent variable forms $\partial Z_\beta/\partial t = (Z_\beta^{n+1}-Z_\beta^n)/\Delta t$, $Z_\beta = (1-\varphi)Z_\beta^n+\varphi Z_\beta^{n+1}$ and $Z_\gamma Z_\beta = (1-\varphi)Z_\gamma^n Z_\beta^n+\varphi Z_\gamma^{n+1}Z_\beta^{n+1}$. Note that $V_{\alpha i}V_{\beta j}$ and $V_{\alpha i}T_\beta$ are assumed to vary linearly with time, which is reasonable if the time step is small. These forms are substituted into equations (81) and (82). After integrating over φ, one divides both equations by $\int_0^1 U(\varphi)d\varphi$ and defines a temporal parameter ς such that:

$$\zeta = \frac{\int_0^1 U(\varphi)\varphi d\varphi}{\int_0^1 U(\varphi)d\varphi} \ . \tag{87}$$

This yields the following linearized algebraic equations for velocity and temperature at $n+1$:

$$\left[M^1_{\alpha\beta} + \zeta \Delta t \left(\begin{array}{c} M^2_{\alpha\gamma\beta j}V^{n+1}_{\gamma j} \\ + M^3_{\alpha\beta} \end{array}\right)\right] V^{n+1}_{\beta i} = \left[M^1_{\alpha\beta} - (1-\zeta)\Delta t \left(\begin{array}{c} M^2_{\alpha\gamma\beta j}V^n_{\gamma j} \\ + M^3_{\alpha\beta} \end{array}\right)\right] V^n_{\beta i} - \Delta t M^4_{\alpha j\beta i}\left(\begin{array}{c} (1-\zeta)V^n_{\beta j} \\ + \zeta V^{n+1}_{\beta j} \end{array}\right)$$

$$- \Delta t M^5_{\alpha\beta i}P_\beta + \Delta t Gra_i M^6_{\alpha\beta}A_{\alpha\beta}\left(\begin{array}{c} (1-\zeta)T^n_\beta \\ + \zeta T^{n+1}_\beta \end{array}\right) + \Delta t G^M_{\alpha i} \ , \tag{88}$$

$$\left[E^1_{\alpha\beta} + \zeta \Delta t \left(\begin{array}{c} E^2_{\alpha\gamma\beta j}V^{n+1}_{\gamma j} \\ + \frac{1}{Pr}E^3_{\alpha\beta} \end{array}\right)\right] T^{n+1}_\beta = \left[E^1_{\alpha\beta} - (1-\zeta)\Delta t \left(\begin{array}{c} E^2_{\alpha\gamma\beta j}V^n_{\gamma j} \\ + \frac{1}{Pr}E^3_{\alpha\beta} \end{array}\right)\right] T^n_\beta + \Delta t G^E_\alpha \ . \tag{89}$$

To employ the Crank-Nicholson method, one sets $U(\varphi) = 1$ which yields $\varsigma = {}^1/2$. Finite element equations for P_d' and V_i' are obtained by setting the spatial residuals in equations (83) and (84) equal to zero.

The surface stress condition is embodied in the Neumann tensor $G_{\alpha i}{}^M$, and is applied along the elements adjoining surface 4 (i.e., the corner and meniscus regions in figure 5). Note that the velocities on all other boundaries are Dirichlet-type. A finite element expression for $G_{\alpha i}{}^M$ is obtained by substituting the tangential stress condition, equation (50), into the definition of $G_{\alpha i}{}^M$ in equation (36):

$$G^M_{\alpha i} = -Re \int_\Gamma \overset{*}{\Phi}_\alpha T_{,i}d\Gamma \ . \tag{90}$$

There are several ways to discretize the temperature gradient along the boundary. One is to assume $T_{,i} = \theta^*_{\beta,i}T_\beta$ and treat θ^*_β as a one-dimensional, second-order interpolation along the element surface. The main advantage with this is that the integral $\int_\Gamma \Phi^*_\alpha\theta^*_{\beta,i}$ is an explicit function of normalized surface position ξ, surface length $l^{(e)}$ and orientation s_i. This approach works well if the surface is horizontal or vertical, and the gradient acts in either the x_1- or x_2-direction. With a curved interface considerable error is introduced because only three temperature values are used to express the gradient in two directions. An alternative approach is to treat $\theta^*_{\beta,i}$ as two-dimensional and include contributions from all element nodes. The integration is carried out while holding the edge corresponding to $\xi_2 = \eta = 1$ constant. First, the discretized form of equation (90) is expressed as a sum of individual element contributions:

$$G^M_{\alpha i} = -Re \sum_{e=1}^{E_4} \overset{*}{\Omega}{}^1_{NMi}T^{(e)}_M \Delta_{N\alpha} \ , \tag{91}$$

where

$$\overset{*}{\Omega}{}^{1}_{NMi} = \int_{\Gamma^{(e)}} \overset{*}{\Phi}{}^{(e)}_{N} \overset{*}{\theta}{}^{(e)}_{M,i} d\Gamma^{(e)} \ . \tag{92}$$

E_4 is the number of elements along surface 4, and $T^{(e)}{}_M$ are local node values of temperature. M is summed over all local nodes. The surface test function $\Phi^{*(e)}{}_N$ is expressed as second-order with respect to ξ:

$$\overset{*}{\Phi}{}^{(e)}_{N} = \frac{1}{2} \begin{bmatrix} \xi(\xi-1) \\ -2(\xi^2-1) \\ \xi(\xi+1) \end{bmatrix} \ . \tag{93}$$

The surface differential $d\Gamma^{(e)}$ is also expressed in terms of ξ by $d\Gamma^{(e)} = (l^{(e)}/2)d\xi$. Note that $\theta^{*(e)}{}_{M,i}$ is merely the derivative of the temperature interpolation $\theta^{(e)}{}_{M,i}$ at $\eta = 1$.

$$\overset{*}{\theta}{}^{(e)}_{M,i} = \theta^{(e)}_{M,i}\Big)_{\eta=1} \ . \tag{94}$$

Equation (92) can now be expressed in the form:

$$\overset{*}{\Omega}{}^{1}_{NMi} = \frac{l^{(e)}}{2} \int_{\xi=-1}^{1} \overset{*}{\Phi}{}^{(e)}_{N}(\xi) \theta^{(e)}_{M,i}\Big)_{\eta=1} d\xi \ . \tag{95}$$

The integration in equation (95) is performed most efficiently by means of one-dimensional Gaussian quadrature where the expression is approximated as:

$$\overset{*}{\Omega}{}^{1}_{NMi} = \frac{l^{(e)}}{2} \sum_{r=1}^{n} w_r \overset{*}{\Phi}{}^{(e)}_{N}(\xi_r) \theta^{(e)}_{M,i}(\xi_r, n=1) \ . \tag{96}$$

w_r is the weight coefficient corresponding to the point index r. $\Phi^{*(e)}{}_N(\xi_r)$ and $\theta^{(e)}{}_{M,i}(\xi_r, \eta=1)$ denote the abscissae evaluated at each of the n Gaussian points. For this summation, a five-point quadrature is again employed, i.e., $n = 5$, since the order of variation for temperature is fourth-order in the corners and surface.

Surface integrals are also used to balance flow through the cavity and specify the Dirichlet condition for V_2 along surface 2. One takes the approach of relaxing the normal stress boundary condition and holding the surface fixed $V_i^{(s)} = 0$. Integration of the scaled expression for mass flux, equation (40), over surface 4 yields the scaled total mass/volumetric flow J (i.e., throughput) in the pore:

$$J = \frac{1}{Rs} \int_{\Gamma_4} T d\Gamma \ . \tag{97}$$

J is approximated by descritizing T in terms of $\theta^{*(e)}{}_M$ and expressing the integral as a summation over surface 4 element contributions, that is:

$$J = \frac{1}{Rs} \sum_{e=1}^{E_4} \overset{*}{\Omega}{}_M^2 T_M^{(e)} \ , \qquad (98)$$

where

$$\overset{*}{\Omega}{}_M^2 = \int_{\Gamma^{(e)}} \overset{*}{\theta}{}_M^{(e)} d\Gamma^{(e)} \ .$$

Along the corner elements, a fourth-order interpolation is applied and the summation occurs over all five surface values of temperature. On the remaining meniscus, a second-order variation is applied to yield:

$$\overset{*}{\Omega}{}_M^2 = \frac{l^{(e)}}{90} \begin{bmatrix} 7 & 32 & 12 & 32 & 7 \end{bmatrix}^T \ , \quad \text{Fourth-Order Variation} \qquad (99)$$

$$\overset{*}{\Omega}{}_M^2 = \frac{l^{(e)}}{6} \begin{bmatrix} 1 & 4 & 1 \end{bmatrix}^T \ . \qquad \text{Second-Order Variation}$$

The rate in equation (97) is set equal to the flux integral on surface 2 by:

$$J = \int_{\Gamma_2} V_2 n_2 d\Gamma \ . \qquad (100)$$

Recall that $V_1 = 0$, consistent with the assumption of parallel flow. In addition, V_2 is treated as a constant (uniform flow) but is assumed to vanish at $x_1 = 0$ and 1 (no-slip). Noting that $n_2 = 1$, V_2 is discretized in terms of $\Phi^{*(e)}{}_M$, and a second expression for J is obtained.

$$J = \frac{\Delta x_1 V_2}{6} \left[10 + 6(E_2 - 2) \right] \ , \qquad (101)$$

where Δx_1 represents the individual element lengths along surface 2, and V_2 is the velocity at each node. Note that E_2 is the number of elements along surface 2. Setting equation (101) equal to equation (98) yields an expression for V_2 which is treated as the Dirichlet boundary condition at each temperature iteration.

$$V_2 = \frac{3J}{\Delta x_1 (3E_2 - 1)} \ . \qquad (102)$$

The Robin boundary condition for temperature, equation (54), is applied using the definition of $G_a{}^E$ in equation (86). The temperature gradient in $G_a{}^E$ is replaced with equation (54) to yield:

$$G_\alpha^E = -\frac{Bi}{Pr} \int_{\Gamma_4} \overset{*}{\theta}{}_\alpha T d\Gamma \ . \qquad (103)$$

Temperature is again discretized by $\theta^{*(e)}{}_M$ to yield a finite element expression for the integral:

$$\int_{\Gamma_4} \overset{*}{\theta}{}_\alpha T d\Gamma_4 = \sum_{e=1}^{E_4} \overset{*}{\Omega}{}_{NM}^3 T_M^{(e)} \Delta_{N\alpha} \ , \qquad (104)$$

37

where

$$\overset{*}{\Omega}{}^{3}_{NM} = \int_{\Gamma_4} \overset{*}{\theta}{}^{(e)}_{N} \overset{*}{\theta}{}^{(e)}_{M} d\Gamma^{e} \ . \tag{105}$$

Removing the temperature gradient in equation (103) eliminates the need to apply an interpolation function based on all element node contributions. Rather, a second- and fourth-order interpolations are applied with respect to the naturalized coordinate ξ_1 coincident with the surface. Integration of the interpolation product in equation (105) then yields the matrices:

$$\overset{*}{\Omega}{}^{3}_{NM} = \frac{l^e}{30} \begin{bmatrix} 4 & 2 & -1 \\ - & 16 & 2 \\ - & - & 4 \end{bmatrix}, \qquad \text{second-order} \tag{106}$$

$$\overset{*}{\Omega}{}^{3}_{NM} = \frac{l^e}{5,670} \begin{bmatrix} 292 & 296 & -174 & 56 & -29 \\ - & 1,792 & -384 & 256 & 56 \\ - & - & 1,872 & -384 & -174 \\ - & - & - & 1,792 & 296 \\ - & - & - & - & 292 \end{bmatrix} . $$

fourth-order

C. Free Surface Solution

With moderate to large contact angles (i.e., $\omega = 15°-90°$), $y^{(s)}$ can be easily calculated using a one-dimensional finite difference or finite element representation of equation (45). Such an approach is difficult with small contact angles because of the high gradient (Neumann condition) at the wall. An alternative is to transform equation (45) into an integral equation by treating α and surface position s as dependent and independent variables, respectively. This approach has been applied to purely hydrostatic configurations,[52 53] and can be extended to include the effects of dynamic pressure, velocity, and interfacial temperature.

First, a new coordinate system (x',y') centered at the base of the meniscus where $x' = x_1-0.5$ and $y' = y^{(s)}-y_0^{(s)}$ is defined. α is related to $y^{(s)}$ by $y^{(s)} = y_0^{(s)} + \int_0^{x'} \tan \alpha dx'$ and $y^{(s)}_{,x} = \tan \alpha$. These definitions are substituted into equation (45) to obtain:

$$\kappa = \frac{\Sigma - C + Bo \int_0^{x'} \tan \alpha \, dx'}{1-D} , \tag{107}$$

$$\Sigma = Boy_0^{(s)} + Ca(P^{(v)} - P_{d0}) + \frac{f_\rho-1}{Rs^2} T_0^2 + 2V_{0i,j} n_j n_i , \tag{108}$$

where

$$C = Ca \left[\Delta P_d - \frac{(f_\rho-1)}{Rs^2} \Delta(T^2) - 2\Delta(V_{i,j}) n_j n_i \right] ,$$

$$D = CrT ,$$

and

$$\Delta P_d = P_d - P_{d0} \ ,$$

$$\Delta(T^2) = T^2 - T_0^2 \ ,$$

$$\Delta(V_{i,j}) = V_{i,j} - V_{0i,j} \ .$$

C and D are derived from the steady-state flowfield solution and are expressed as functions of x'. Between the surface nodes at which flowfield data is available, T, V_i, and P_d in equation (108) are interpolated using the same order as their element-level variance. Equation (107) is expressed in a form where Σ corresponds to the curvature at the meniscus center-line. Consequently, the curvature at other points along the surface is a function of Σ and the change in pressure, temperature and velocity relative to the center-line values, T_0, V_{0i}, and P_{d0}.

For a two-dimensional meniscus, only the radius of curvature κ in the x'-y' plane is considered. The well-known expression for κ in this case[53] is:

$$\kappa = -\frac{1}{y_x^{(s)} dx} \frac{d}{dx} \left\{ \left[1 + (y_x^{(s)})^2 \right]^{-1/2} \right\} \ . \tag{109}$$

Upon substitution with the expressions for $y^{(s)}$ and $y^{(s)}_{,x}$, equation (109) becomes $\kappa = \cos \alpha (\partial \alpha / \partial x)$. For moderate to large contact angles, this equation can be used with equation (107) to solve α versus x' using a combined finite difference-integral expression. Convergence problems occur at low contact angles due to the rapid increase in the tangent term near the wall. This problem is corrected by using the surface distance s, rather than x', as the independent variable. s is introduced into the formulation by noting that $dx'/ds = \cos \alpha$ and $dy'/ds = \sin \alpha$. Substituting these expressions into the two α-dependent terms in equation (107) yields $\kappa = \partial \alpha / \partial s$ and $\int_0^{x'} \tan \alpha dx' = \int_0^s \sin \alpha ds$. The transformed expression for equation (107) becomes:

$$\frac{\partial \alpha}{\partial s} = \frac{\Sigma - C + Bo \int_0^s \sin \alpha ds}{1 - D} \ . \tag{110}$$

Equation (110) is expressed in numerical form by applying difference and integral approximations for $\partial \alpha / \partial s$ and $\int_0^s \sin \alpha ds$, respectively. The entire surface extending from $x' = -0.5$ to $x' = 0.5$ is solved by applying the shooting method to each half of the meniscus. The objective is to determine a Σ that yields the specified contact angle ω at $|x'| = 0.5$. At $x' = 0$, $\alpha = 0$ is set, a value of Σ is assumed, and α_i is calculated at successive steps along the contour until $|x'| \geq 0.5$. A first-order approximation for $\partial \alpha / \partial s$ suffices in flat regions of the meniscus, but becomes less accurate towards the walls. Computational precision in this region is improved by making the step size $\Delta s = s_i - s_{i-1}$ extremely small ($\sim 10^{-4}$) and by applying Simpson's rule to obtain second-order approximations of the derivative and integral. The first and second-order accurate forms of $\partial \alpha / \partial s$ are:

$$\frac{\partial \alpha}{\partial s} \approx \frac{\alpha_i - \alpha_{i-1}}{s_i - s_{i-1}} \ , \qquad \text{first-order} \tag{111}$$

$$\frac{\partial \alpha}{\partial s} \approx \frac{3\alpha_i - 4\alpha_{i-1} + \alpha_{i-2}}{2\Delta s} \ , \qquad \text{second-order}$$

where α_i and α_{i-1} are the values of α at s_i and s_{i-1}, respectively. Also, a backward difference form of $\partial\alpha/\partial s$ has been used for the second-order expression. Noting that $\int_0^s \sin\alpha\,ds$ in equation (110) represents the y' value at step i (i.e., y'_i), the integral is expressed as $\int_0^s \sin\alpha\,ds = y'_{i-1} + \Delta y'_i$, where y'_{i-1} is the y' value calculated from the previous iteration and $\int_0^s \sin\alpha\,ds$ is the value of the integral from $i-1$ to i. The trapezoid and Simpson's rules are applied to derive first- and second-order approximations for $\Delta y'_i$, respectively:

$$\Delta y'_i = \frac{(s_i - s_{i-1})}{2} (\sin\alpha_{i-1} + \sin\alpha_i) , \qquad \text{first-order} \qquad (112)$$

$$\Delta y'_i = \frac{\Delta s}{6} (\sin\alpha_i + 4\sin\bar{\alpha} + \sin\alpha_{i-1}) . \qquad \text{second-order}$$

y'_{i-1} is the sum of the interval integrals calculated from previous iterations:

$$y'_{i-1} = \sum_{n=2}^{i-1} \Delta y'_n . \qquad (113)$$

The angle $\bar{\alpha}$ in equation (112) is the value of α evaluated midway between s_i and s_{i-1}. It can be expressed in terms of the angles evaluated at i, $i-1$ and $i-2$ by expanding $\bar{\alpha}$ into a Taylor series about α_{i-1}. Ignore terms higher than second-order and approximate the derivatives using center differences about α_{i-1} to yield a second-order accurate expression for $\bar{\alpha}$:

$$\bar{\alpha} \approx \frac{1}{8} (3\alpha_i + 6\alpha_{i-1} - \alpha_{i-2}) . \qquad (114)$$

Substitution of equations (111), (112), and (114), into equation (110) yields the following nonlinear equations for α_i at each contour integration step:

$$\alpha_i = \alpha_{i-1} + \frac{1}{1-D}\left[\begin{array}{l} (\Sigma - C + Bo\,y'_{i-1})\,\Delta s \\ + \frac{Bo\Delta s^2}{2}(\sin\alpha_{i-1} + \sin\alpha_i) \end{array} \right], \qquad \text{first-order} \qquad (115)$$

$$\alpha_i = \frac{4\alpha_{i-1} - \alpha_{i-2}}{3} + \frac{1}{3(1-D)}\left[\begin{array}{l} 2(\Sigma - C + Bo\,y'_{i-1})\Delta s \\ + \frac{Bo\Delta s^2}{3}(\sin\alpha_{i-1} + 4\sin\bar{\alpha} + \sin\alpha_i) \end{array} \right]. \qquad \text{second-order}$$

After each iteration, x'_i is calculated from:

$$x'_i = x'_{i-1} + \frac{\Delta s}{2}(\cos\alpha_i + \cos\alpha_{i-1}) , \qquad \text{first-order} \qquad (116)$$

$$x'_i = x'_{i-1} + \frac{\Delta s}{6}(\cos\alpha_i + 4\cos\bar{\alpha} + \cos\alpha_{i-1}) . \qquad \text{second-order}$$

40

In most calculations, the final x'_i-estimate overshoots the length constraint (i.e., $|x'| > 0.5$). A cubic approximation of s_i versus x'_i based on s_i, s_{i-1}, s_{i-2}, and s_{i-3} is used to interpolate a final Δs_i that yields $|x'| = 0.5$. This value of Δs_i is used to determine a final α_i and contact angle estimate $\omega_i = \pi/2 - \alpha_i$. If ω_i fails to match, within a specified tolerance, the desired value of ω, Σ is adjusted, and the contour integration is repeated. If $\omega_i > \omega$, then Σ is increased, while if $\omega_i < \omega$, it is decreased. The new estimate of Σ is obtained using a simple bisection algorithm, which enables accommodation of large positive and negative curvatures at the center-line.

In the actual procedure, one only needs to calculate one side of the meniscus since the geometry and flowfield are symmetric about $x' = 0$. However, the same general approach could be applied to unsymmetrical problems by shooting the solution to the desired contact angle and treating Σ and the final x'_i-estimate as the independent and dependent variables, respectively. The iteration would continue until the sum of x'_i-estimates on both sides were equal to 1.0. The meniscus center would then be adjusted until the final x'_i-estimates coincided with the side walls.

D. Summary

The equations and algorithms outlined in this section serve as the basis for the Combined Interfacial Convection (CIC) computer program, which was used to obtain the results in sections VI, VII, and VIII. A brief description of the program, including its input/output format, subprogram structure, key variables and general capabilities, is provided in appendix B. A complete listing of the CIC program module and its associated subprogram elements is given in appendix C.

In the CIC code, calculation of meniscus geometry is the outermost iteration in a five-loop flowfield solution procedure. The first (i.e., innermost) loop determines a velocity field that satisfies equation (88) based on an estimated pressure distribution. The second loop (i.e., SIMPLER algorithm) adjusts the velocity and pressure field by solving equations (83) to (85) in which $S_\alpha \Delta P1 = S_{\alpha i} \Delta V = S_\alpha \Delta P2 = 0$. Once convergence is established for V_i and P_d, the third loop computes the temperature field using equation (89). In the fourth loop, the time step is advanced to steady-state, and temporal convergence is checked. The criterion is that the variables vary no more than 0.01 percent from their values at the previous time step.

In the velocity and SIMPLER loops, the dynamic pressure is fixed $P_d = 0$ at the two corners at the base of the pore (i.e., lower left and right-hand corners). By separating pressure from the stress tensor and restricting application of Green's theorem to the viscous stress tensor in the weak formulation of the momentum equation, one eliminates the need to specify pressure at the other boundary nodes.

V. SCALING ANALYSIS

A. Introduction

The finite element equations derived in section IV provide the basis for determining the velocity, pressure, and temperature within the domain in figure 4. To make such an investigation physically relevant, one ascertains via parametrics the magnitude and sensitivity of the principal dimensionless groupings identified in section III over a range of thermophysical properties, pore widths D, and superheating/subcooling limits $|\Delta T|$. This is necessary in order to identify the parameter ranges representative of pores with $1 \ \mu m < D < 10^2 \ \mu m$ and $10^{-1} \ K < |\Delta T| < 1 \ K$. Using the properties for four fluids commonly used in capillary systems as a reference (table 3), a modified parameter value, in which D and $|\Delta T|$ have been factored out, is calculated for each fluid and compiled in table 4. The minimum and maximum values for each modified parameter define an approximate order of magnitude range for the thermophysical variance of each dimensionless grouping.

To evaluate parameter sensitivity to pore size, one considers a range of D from $10^{-2} \ \mu m$ to $10^4 \ \mu m$. Although $1 \ \mu m$ represents a reasonable lower limit for most fine mesh screens, porous surfaces and wicking structures, $10^{-2} \ \mu m$ is selected to broaden the range of study. In addition, $10^{-2} \ \mu m$ generally represents the upper limit at which long-range intermolecular forces between the liquid and solid begin to influence surface morphology. It is assumed that these forces can be accounted for on the macroscopic-level by assuming a fixed contact angle between the meniscus and side wall. The upper bound of D, $10^4 \ \mu m$ (= 1 cm), is arbitrarily defined as a maximum, since characteristic dimensions in which $D \geq 10^5 \ \mu m$ (= 10 cm) fall more in the category of a small container and out of the size range of this study.

Table 3. Fluid properties.

Property	H$_2$O	Ethanol	Ammonia	R-114
T_{sat} (K)	373.1	351.7	239.8	276.9
ρ_l (kg/m^3)	9.59×10^2	7.89×10^2	6.83×10^2	1.52×10^3
ρ_v (kg/m^3)	5.98×10^{-1}	1.49	8.90×10^{-1}	7.83
μ_l (kg/(m-s))	2.83×10^{-4}	3.72×10^{-4}	2.69×10^{-4}	5.06×10^{-4}
μ_v (kg/(m-s))	1.29×10^{-5}	1.08×10^{-5}	9.00×10^{-6}	1.04×10^{-5}
k_l (W/(m-K))	6.82×10^{-1}	1.71×10^{-1}	5.48×10^{-1}	9.23×10^{-2}
k_v (W/(m-K))	2.51×10^{-2}	2.30×10^{-2}	2.13×10^{-2}	9.84×10^{-3}
Cp_l (J/(kg-K))	4.21×10^3	2.44×10^3	4.47×10^3	1.00×10^3
Cp_v (J/(kg-K))	1.89×10^3	1.69×10^3	2.18×10^3	6.34×10^2
α_l (m^2/s)	1.69×10^{-7}	8.88×10^{-8}	1.79×10^{-7}	6.05×10^{-8}
α_v (m^2/s)	2.22×10^{-5}	9.16×10^{-6}	1.10×10^{-5}	1.98×10^{-6}
L (J/kg)	2.26×10^6	8.79×10^5	1.37×10^6	1.36×10^5
M_w (kg/kgmole)	18.02	46.07	17.03	170.94
γ (N/m)	5.88×10^{-2}	2.00×10^{-2}	4.24×10^{-2}	2.57×10^{-2}
$|d\gamma/dT|$ (N/(m-K))	1.94×10^{-4}	9.69×10^{-4}	3.26×10^{-4}	1.15×10^{-4}
β (1/K)	7.72×10^{-4}	1.10×10^{-3}	2.45×10^{-3}	1.78×10^{-3}

Values at 0.1013 MPa.

Table 4. Dimensionless groupings—factored form.

Parameter	H_2O	Ethanol	Ammonia	R-114	Range
$f\rho$	1.60×10^3	5.30×10^2	7.68×10^2	1.94×10^2	$10^2 \leftrightarrow 10^3$
f_k	3.68×10^{-2}	1.35×10^{-1}	3.89×10^{-2}	1.07×10^{-1}	$10^{-2} \leftrightarrow 10^{-1}$
f_μ	4.58×10^{-2}	2.90×10^{-2}	3.35×10^{-2}	2.04×10^{-2}	$10^{-2} \leftrightarrow 10^{-1}$
Pr (liquid)	1.75	5.31	2.19	5.51	$1 \leftrightarrow 10$
λ	2.34×10^{-1}	2.27×10^{-2}	3.19×10^{-1}	3.97×10^{-1}	$10^{-2} \leftrightarrow 1$
$Ma/(D\Delta T)$	4.06×10^6	2.93×10^7	6.76×10^6	3.76×10^6	$10^6 \leftrightarrow 10^8$
$Bo/(D^2 a)$	1.60×10^5	3.87×10^5	1.58×10^5	5.81×10^5	$10^5 \leftrightarrow 10^6$
$Bi_c/(Dh)$	1.47	5.85	1.82	1.08×10^1	$1 \leftrightarrow 10$
Bi_e/D	1.15×10^7	3.03×10^7	1.48×10^7	1.96×10^7	$10^7 \leftrightarrow 10^8$
$RsD\Delta T$	8.13×10^{-5}	6.31×10^{-5}	4.54×10^{-5}	3.82×10^{-5}	$10^{-5} \leftrightarrow 10^{-4}$
$Cr/\Delta T$	3.29×10^{-3}	4.84×10^{-2}	7.69×10^{-3}	4.48×10^{-3}	$10^{-3} \leftrightarrow 10^{-1}$
A/D^2	2.60×10^{19}	3.96×10^{18}	8.85×10^{18}	1.23×10^{18}	$10^{18} \leftrightarrow 10^{19}$
$Bi_{ke}/(D^3 \Delta T^2)$	8.63×10^1	2.70×10^2	2.39×10^2	2.05×10^2	$10^2 \leftrightarrow 10^3$
$Vr/(D\Delta T^2)$	3.44×10^2	1.16×10^3	9.29×10^2	8.68×10^2	$10^2 \leftrightarrow 10^3$
$Gr/(aD^3 \Delta T)$	8.72×10^{10}	4.86×10^{10}	1.61×10^{12}	1.57×10^{11}	$10^{10} \leftrightarrow 10^{11}$
CaD	1.42×10^{-9}	8.77×10^{-9}	2.50×10^{-9}	6.57×10^{-9}	$10^{-9} \leftrightarrow 10^{-8}$

The maximum superheat ΔT that can be sustained without causing nucleation depends on the fluid, contact surface roughness, and pore width. For the size range considered here, a bound of 10^{-1} K to 1 K seems reasonable. However, to emphasize the effect of larger superheats, which are encountered with thin films, an upper limit of 10 K is assumed.

B. Meniscus Solution

First, the parameters arising from equation (45), namely Bo, Vr, Cr, and Ca are examined. For the purely static, saturated case ($P_d = V_i = T = 0$), the terms containing Ca, Vr, and Cr vanish leaving an expression dependent solely on Bo. The sensitivity of surface morphology for this case is illustrated in figure 7 which shows the left half of the meniscus surface for different values of Bo. The lower limit, $Bo \rightarrow 0$, yields a constant-curvature interface, which is virtually circular for $Bo < 10^{-1}$. With larger Bo, the meniscus must flatten at the center-line to accommodate the increasing curvature towards the side walls. At the upper limit, $Bo \rightarrow \infty$, the interface becomes flat, and curvature in the interline region is practically indiscernible.

Heating or cooling establishes a temperature and velocity distribution along the surface, and requires the consideration of dynamic pressure, vapor recoil, and surface tension variation to solve equation (45) and its numerical counterpart, equation (110). Because the flow variables are scaled with viscous timescales, the influence of dynamic pressure is dictated by Ca which relates viscous stress to nominal surface tension. The greatest variation in P_d occurs in regions of high velocity gradient, such as the side wall and meniscus interline. A sharp change in P_d is expected at the interline followed by a somewhat constant distribution in the center of the pore. Since the value of P_d can be either positive or negative, the surface P_d distribution may either offset or reinforce the
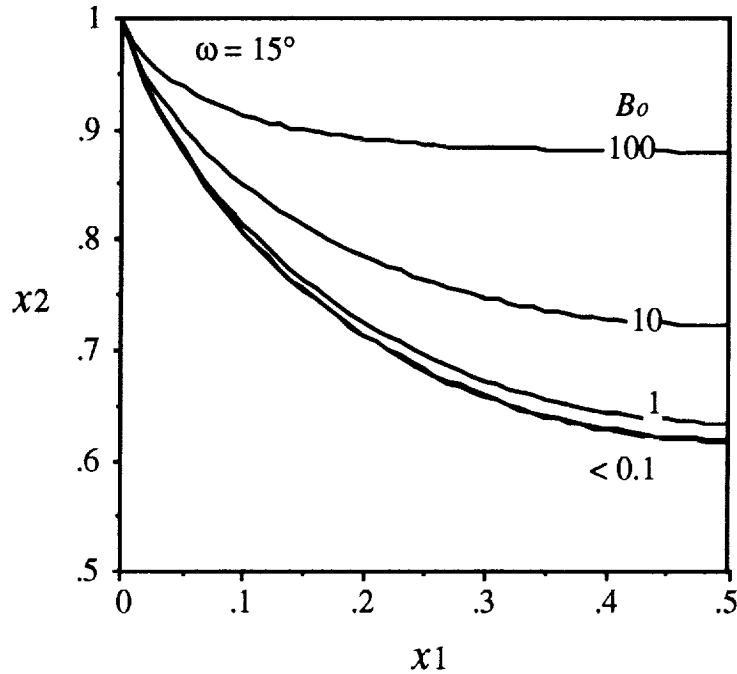
Figure 7. Meniscus geometry for static isothermal pore.

influence of Bo on curvature. From a physical standpoint, a positive P_d in the x'-direction would oppose action of the imposed acceleration and hydrostatic pressure, causing higher curvatures in the middle of the meniscus. The converse situation would hold if the pressure gradient was negative.

The effect of vapor recoil is embodied in the term containing Vr. The term is always positive, since $Vr > 0$ and $T^2 > 0$. This indicates that the recoil force from vapor ejection or impingement always acts in the $-x_2$-direction and augments the contribution of hydrostatic pressure on curvature. Because the recoil term is proportional to T^2, this effect is more pronounced at the side walls, where the scaled temperature $|T|$ is highest. To accommodate the increasing curvature towards the side wall, a lower curvature is required in the center to maintain equilibrium. This causes the surface near the side wall to appear depressed relative to the center. The magnitude of this depression depends on Vr and the surface temperature distribution. It is also possible that for high values of Vr the equilibrium interface may assume a negative curvature and bulge upwards in the middle of the surface.

Thermally induced surface tension variation is represented by the denominator of equation (45). Cr arises from expressing γ as a linear function of temperature. Although Cr is ordinarily quite low up to moderate levels of superheating/subcooling, at high $|\Delta T|$'s it can have an appreciable effect. For a typical fluid ($0 < Cr < 1$), heating ($T > 0$) decreases surface tension and causes an increase in curvature towards the side wall, similar to the recoil term. At higher temperatures, the curvature must increase to preserve equilibrium and offset reduced surface tension. With cooling ($T < 0$), the opposite trend occurs and the curvature is maximized at the center, which is manifested as a surface depression in the middle of the cavity.

Plots of the upper and lower thermophysical ranges of Bo, Cr, Vr, and Ca versus D for $|\Delta T|$'s of 10^{-1} K and 10 K are shown in figure 8.
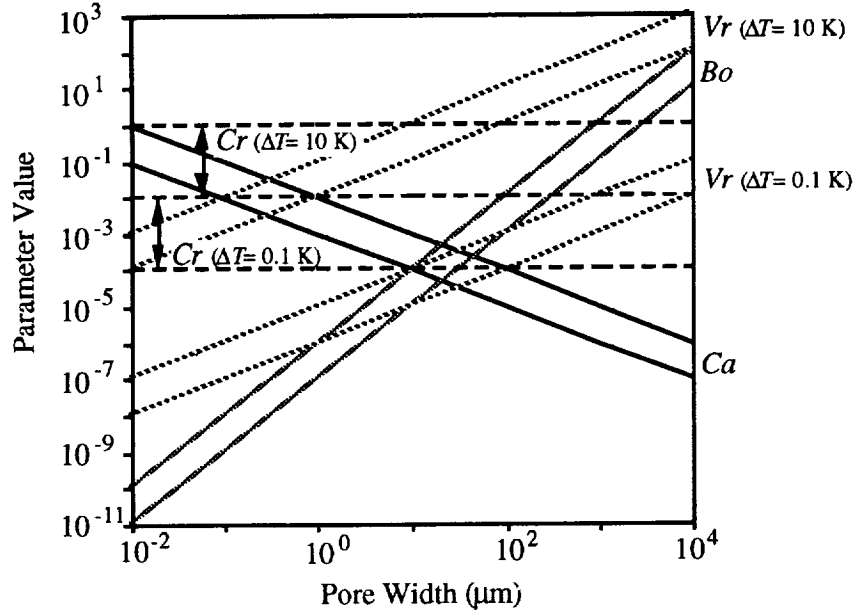
Figure 8. Normal stress-related parameters.

Bo and *Ca*, which characterize the influence of hydrostatic pressure, viscosity, and bulk surface tension, are independent of |Δ*T*| and exhibit nearly opposite functional dependence on *D* (i.e., $Bo \propto D^2$ and $Ca \propto D^{-1}$). This implies the existence of a capillary-dominated regime at small characteristic dimensions (*D* ~ 1 μm) where *Ca* is large relative to *Bo* and the velocity and dynamic pressure terms in equation (45) influence surface morphology. Depending on the value of superheat, *Vr* and *Cr* may also affect the geometry. Although *Vr* is proportional to *Ca*, it also varies inversely with the square of *Rs* (where $Rs \propto 1/D$) which means that $Vr \propto D$. At low Δ*T*, both *Vr* and *Cr* have a minimal effect on morphology in the capillary regime. However at higher superheats (Δ*T* = 10 K), both can overwhelm viscous effects and dictate interface behavior.

At higher characteristic dimensions (*D* ~ 10^4 μm), *Bo* predominates, and one can effectively ignore contributions from dynamic pressure. At low superheats, the effect of vapor recoil and surface tension variation are also small, and one expects the meniscus to be governed entirely by hydrostatic pressure. At higher superheats, the effect of vapor recoil becomes extremely important and can compete with hydrostatic effects. Since *Cr* < 1, its contribution will be small for large *D*.

C. Velocity/Pressure Solution

Solution of velocity and dynamic pressure requires the parameters in equations (36), (50), and (52). By recasting *Re* in terms of *Ma* and *Pr*, and fixing *Pr* = 1, one obtains the three parameters *Gr*, *Ma*, and *Rs*. *Gr* represents the sensitivity of density to temperature and dictates the magnitude of buoyancy, while *Ma* represents the sensitivity of surface tension to temperature and dictates the magnitude of thermocapillary stress. *Rs* arises from the constitutive expression relating mass flux and surface normal velocity to temperature. To maintain consistency with the definitions of *Ma* and *Gr*, the sensitivity of interfacial mass transfer to temperature is expressed in terms of 1/*Rs*.

The sensitivity of *Gr*, *Ma*, and 1/*Rs* to *D* and Δ*T* are shown in figure 9. Note that these groupings exhibit the same proportional functional dependence on Δ*T*. One finds that the different

functional behaviors of Ma and Gr on D divide the velocity solution into three types of regimes. The thermocapillary regime occurs at $D < 10^3$ μm where $Ma \gg Gr$. In this region, velocity is dictated by surface tension variation and, to a comparable extent, the sensitivity of $1/Rs$ to temperature. The influence of Gr (which follows the same functional behavior as Bo) is essentially nonexistent. Note that $1/Rs$ and Ma exhibit parallel trends with respect to D and ΔT, which would confirm the competition observed by Chen et al.[41]
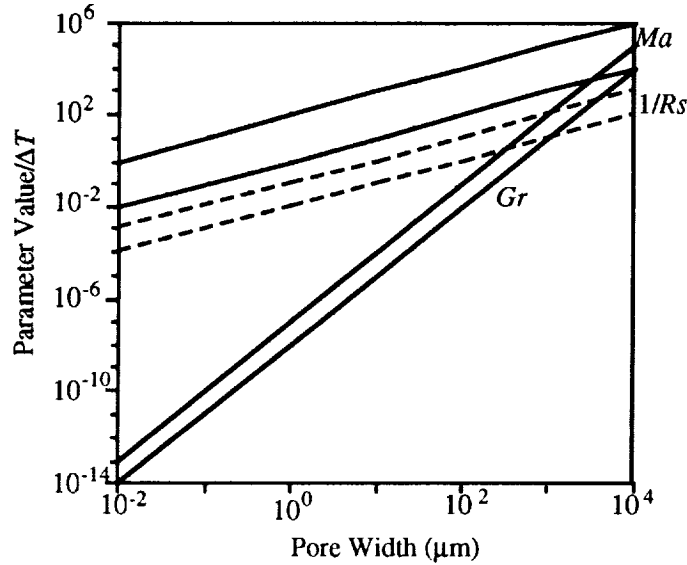
Figure 9. Momentum-related parameters.

Extrapolating the trends in figure 9 to larger pore widths indicates that the regime where buoyancy forces predominate over thermocapillary and evaporative effects occurs at $D > 10^5$ μm. Between this point and the upper limit for pure thermocapillary flow ($D < 10^3$ μm), there exists a regime where buoyancy and thermocapillary forces compete. Here, the forces can either augment or offset each other depending on the orientation of the acceleration field and locations of applied heating and cooling.

D. Temperature Solution

The complete temperature solution primarily involves the boundary condition, equation (54). For the case of two isothermal side walls, one expects a large change in temperature near the side wall followed by a more uniform distribution in the center of the pore with higher Bi_m. At extremely high Bi_m, the scaled temperature only a short distance away from the wall will be negligible (i.e., $T \approx 0$), and thermocapillary effects will concentrate at the walls in a region where there is a notable temperature change along the surface.

The relative sensitivity of the two parameters in equation (54), Bi_m and Bi_{ke}, is shown in figure 10. Bi_m increases linearly with D from 10 at $D = 1$ μm to 10^6 at $D = 10^4$ μm. The functional variation of Bi_{ke} with respect to D and ΔT appears to support the original assumption of ignoring this term in the Neumann condition for temperature. Although Bi_{ke} varies linearly with D^2 and ΔT, its magnitude is so low that it never contributes to interfacial temperature. The trend in figure 10 suggests that it may become important only at characteristic dimensions significantly greater than the upper limit of this study.
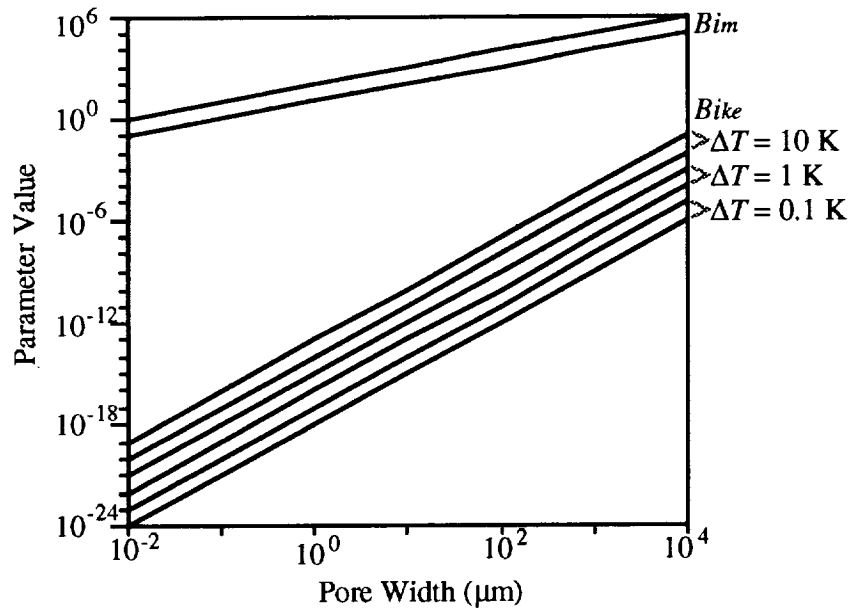
Figure 10. Energy-related parameters.

E. Summary

In this section, the parameter ranges appropriate for studying 1 μm to 10^2 μm wide pores subjected to superheating/subcooling levels between 10^{-1} K and 1 K have been established. It is seen that Gr and Bo become vanishingly small in this size range which implies that buoyancy and hydrostatic effects can be ignored in the problem. The key parameters needed for solution of velocity are Ma and Rs which assume the ranges: $10^{-1} < Ma < 10^3$ and $10^{-1} < Rs < 10^3$. For calculation of the meniscus surface, Cr can range from 10^{-4} for the low value of superheat/subcooling to 10^{-1} at 1 K. A contribution is also expected from Ca which varies from 10^{-2} to 10^{-5}. For temperature, one sees that Bi can vary from 1 to 10^4 over this range.

VI. PORE FLOWFIELD—FIXED SURFACE

A. Introduction

The CIC program described in appendix B and listed in appendix C is first used to investigate the flow and temperature fields in the vicinity of a fixed circular meniscus. Evaluating such a domain is equivalent to considering a gravity-free environment, and ignoring the velocity, pressure, and temperature terms in the normal stress balance. In a one-g environment, this assumption is probably inappropriate, since the pore length scales for which Cr and Ca vanish ($D \geq 10^2$ μm) yield increasing contributions from Bo and hydrostatic pressure. However, in microgravity with moderate dimensions, one would expect to find situations in which such an assumption was valid.

Investigation of a nondeforming domain is useful for understanding how Ma, Rs, and Bi mutually influence convection, heat and mass transfer, thermocapillary stress, interfacial temperature, and circulation. The effect of contact angle ω should also be examined because, as

47

Kamotani and Platt[38] showed, highly wetting liquids incur different stress distributions and circulation intensities than a flat surface. Obtaining reference solutions for interfacial temperature, pressure, and velocity also enables evaluation of the first-order influence of Ma, Rs, Bi, Ca, and Cr on surface morphology.

In this section, four regimes are examined: (1) the basic state, (2) pure evaporation and condensation, (3) subcooled and superheated thermocapillary flow, and (4) combined mode convection. Although liquid motion is ignored in the basic state, this case is important since it reveals the underlying influence of Bi on domain and interfacial temperature. Since the boundary conditions for surface velocity and thermocapillary stress are expressed as functions of T and $\partial T/\partial s$, the basic state interfacial temperature yields insight regarding the driving mechanisms for convection within the cavity. Furthermore, the temperature distributions of the three convective regimes may be viewed as departures from the basic state.

With pure evaporation and condensation, the tangential stress is assumed to be vanishingly small by setting $Ma = 0$. The entire velocity field is driven by the mass flux across the meniscus which is dictated by Rs and surface temperature. In the case of pure thermocapillary flow, the resistance is set to a large value ($Rs = 10^3$) that effectively removes the normal component of velocity at the interface. The flow structure is dictated by Ma and the surface temperature distribution. In combined flow, one examines the simultaneous effect of Ma, Rs, and Bi, where $Ma > 0$ and $Rs < 10^3$.

B. Basic State

In the basic state, heat transfer between the pore boundaries occurs solely by conduction. To examine this regime, one sets $V_i = 0$ and solves the steady-state diffusion equation for temperature $T_{,jj} = 0$ while applying equation (54) on the meniscus. The temperature distribution is a function of Bi and the geometry of the upper boundary which, for a circular meniscus, depends exclusively on contact angle. When applied within this context, Bi still delimits the energy transport within the liquid, but is defined according to the general form in equation (55).

The sensitivity of liquid temperature to Bi is illustrated in figure 11 which shows half-cavity isotherms corresponding to $\omega = 15°$, and $Bi = 1$, 10, and 10^2. Only the left side of the cavity is shown since the distribution is symmetric about the $x_1 = 0.5$ axis. Also, the isotherms are expressed in terms of $|T|$ because the scaled temperatures for superheating and subcooling are equivalent in magnitude and differ only in sign (i.e., $T < 0$ for subcooling and $T > 0$ for superheating). The most obvious trend from these plots is the increase in cavity thermal gradient at larger values of Bi. As $Bi \to \infty$, conduction becomes the rate limiting process for heat transfer, and the gradient must increase to support enhanced heat transfer between the liquid and vapor. Alternatively, as $Bi \to 0$, surface convection becomes the limiting factor, thus reducing the required thermal gradient.

An important consequence of these trends is the role that Bi plays in dictating interfacial temperature. The temperature profiles corresponding to the cases in figure 11 are shown in figure 12. Although the influence of T on convection is conveyed at the boundary by defining $T = T(s)$, since $s = s(x_1)$ at a fixed ω, temperature can just as easily be expressed in terms of x_1. In all cases, the maximum temperature difference between the liquid and vapor is located at $x_1 = 0$ and 1, while the minimum occurs at $x_1 = 0.5$. The difference between the scaled temperatures at the side wall and meniscus center-line is indicative of the average surface temperature gradient.

48
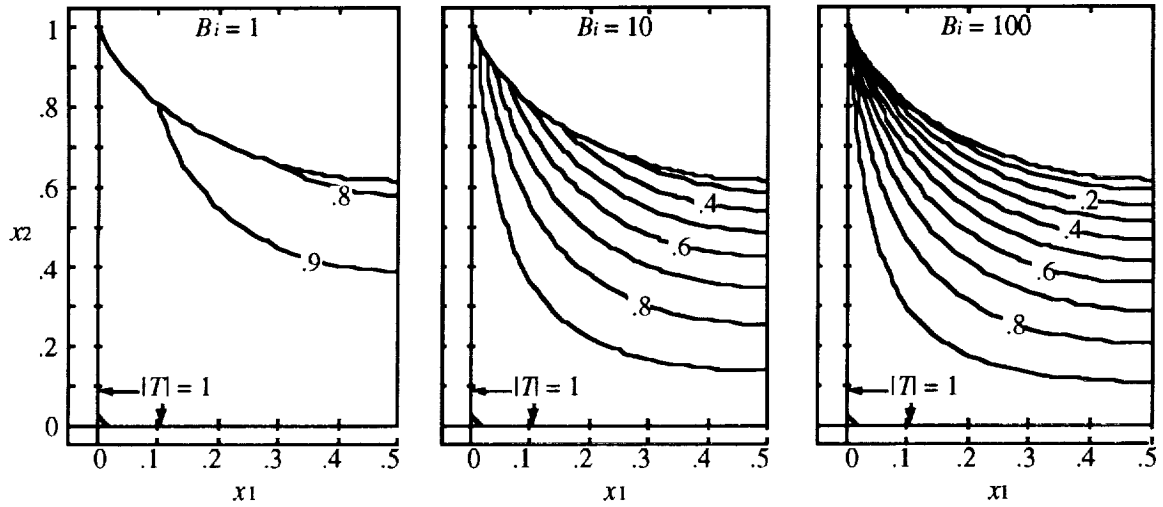
Figure 11. Basic state half-cavity isotherms.
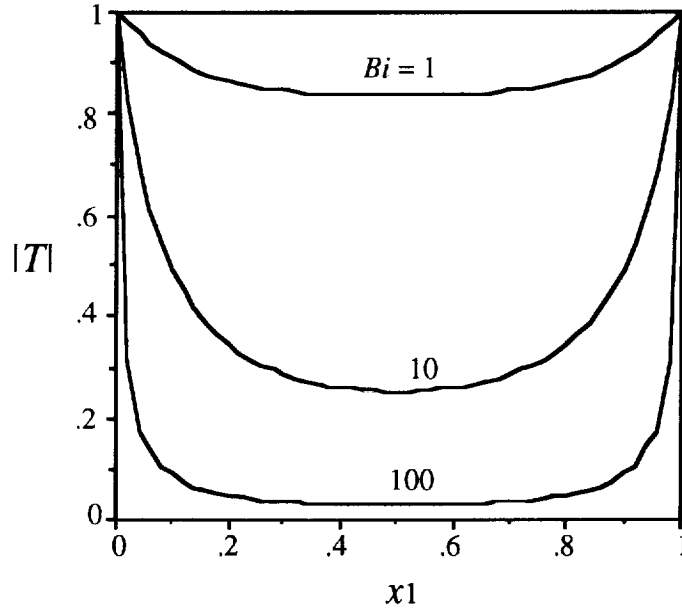
Figure 12. Basic state surface temperature.

In the convection-limited regime represented by low Bi, $|T|$ and $|\partial T/\partial s|$ decrease uniformly towards the center. For $Bi \leq 10$, the effects of conduction and surface convection are nearly equivalent, and the temperature distributions are similar in that the change in $|\partial T/\partial x_1|$ with x_1 is approximately constant. As $Bi \to \infty$, however, the change in $|T|$ and $|\partial T/\partial s|$ is greater which reflects the higher thermal gradient in the domain. For $Bi \geq 10^2$, heat transfer becomes conduction-limited, and $T(x_1)$ exhibits two distinct regions. At the interline, there is a steep temperature gradient followed by a relatively uniform distribution in the middle of the cavity.

Although the interest is in liquids with small contact angles, examining the influence of larger ω is crucial for understanding the unique stress characteristics of highly wetting surfaces. The most significant effects of contact angle are the change in surface area available for heat transfer, and the

variation of T and $\partial T/\partial s$ along the surface. This is evident in figure 13 which shows the temperature magnitude as a function of contour position s and ω with $Bi = 1$, 10, and 10^2 in cavity-centered coordinates. It is clear that the overall change in T and $\partial T/\partial s$ is less with smaller contact angles because of the increase in area and the surface's steeper orientation relative to the internal temperature gradient. As $\omega \to 0$, the total supportable heat transfer becomes greater due to the increase in surface area. Hence, the liquid/vapor temperature difference and gradient along the surface must be less to preserve the pore energy balance.
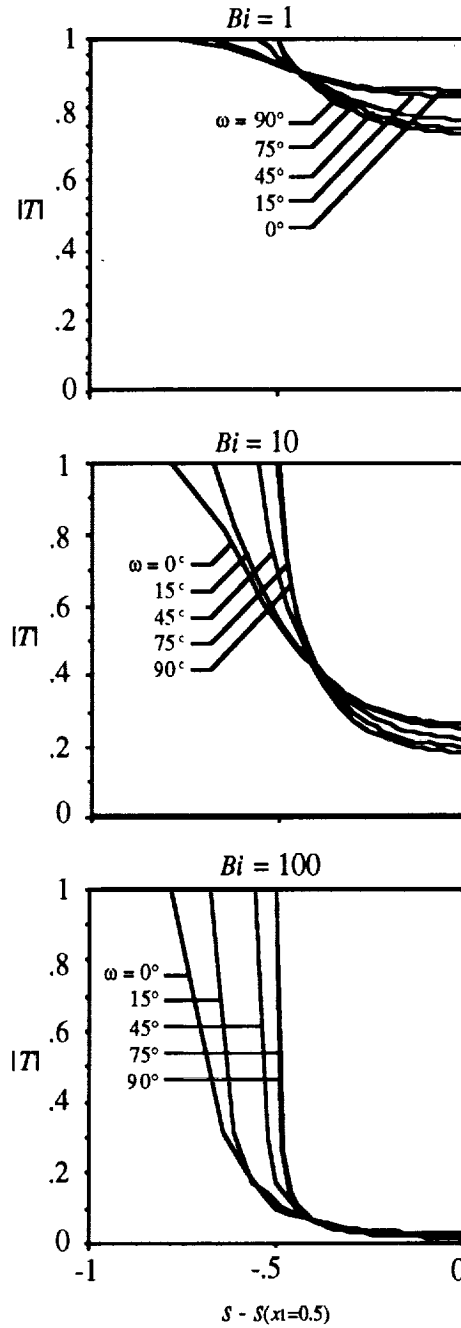
Figure 13. Surface temperature versus contour position and ω.

The surface's steeper orientation relative to the temperature field is exhibited by the negative curvature in $|T(s)|$ near the side wall for $Bi = 1$ and $\omega = 0$. In the contact region, the surface tangent vector s_i points in a direction that is nearly vertical and parallel to the side wall. The temperature along the contour decreases much more slowly with respect to s than the case of a flat meniscus. As $\omega \rightarrow 90°$, $|\partial T/\partial s|$ tends to increase due to the more perpendicular orientation of the interface relative to the side wall.

The sensitivity of $|T|$ to ω, especially in the middle of the meniscus, is greater at lower values of Bi. Temperature variation along the meniscus is continuous and sensitive to the relative orientation between the temperature gradient and surface. As $Bi \rightarrow \infty$, the gradient is shifted toward the side wall, thus diminishing the sensitivity in the center of the cavity. At $Bi = 10^2$, the differences in $|T(s)|$ are primarily due to the variation in $|\partial s/\partial x_1|$ near the side wall, and the temperature distributions are essentially equivalent in the middle of the cavity.

A dimensionless expression for total heat transfer Q is obtained by substituting the scaling factors in equation (33) into the equation for heat flux q, equation (54). Integrating the resulting expression over the meniscus area yields:

$$Q = Bi\Pi , \tag{117}$$

where

$$\Pi = \int_\Gamma T d\Gamma .$$

Π is termed the thermal potential and represents the nonlinear dependency of Q on Bi and ω. With superheating, $Q > 0$ which implies heat transport to the vapor, while with subcooling, heat transfer is directed into the liquid ($Q < 0$). Since fluid convection is ignored, the magnitude of Q and Π are the same for both heating modes. With the temperature distributions shown in figure 11, the values of Π corresponding to $Bi = 1$, 10, and 10^2 are 1.17, 0.60, and 0.16, respectively. Although Π decreases as $Bi \rightarrow \infty$ due to the reduced temperature of the interface, the linear contribution of Bi in equation (117) offsets this decrease and yields corresponding Q values of 1.17, 5.98, and 16.43, respectively. Q and Π exhibit opposite behavior, since Π is related to the average interfacial temperature by $T_{avg} = \Pi/\Gamma$. This is evident from the values for T_{avg} which are 0.86, 0.44, and 0.12 for $Bi = 1$, 10, and 10^2, respectively.

The utility of Π extends to more than meniscus heat transfer and temperature. It is particularly useful for comparing the effects of convection on mass flow through the cavity. With a fixed surface, for example, the total mass flowrate (i.e., throughput) is obtained by integrating equation (40) over the surface to yield:

$$\int_\Gamma j d\Gamma = \frac{\Pi}{Rs} . \tag{118}$$

Π serves a similar role as before by separating out the nonlinear influence of Bi and ω on mass transfer.

The sensitivity of the magnitude of the basic state thermal potential $|\Pi_b|$ to contact angle and Bi is shown in figure 14. As $Bi \rightarrow \infty$, $|\Pi_b| \rightarrow 0$ and the surface approaches the vapor temperature $T \rightarrow 0$. Consistent with the definition of Rs, this represents the state of complete equilibrium
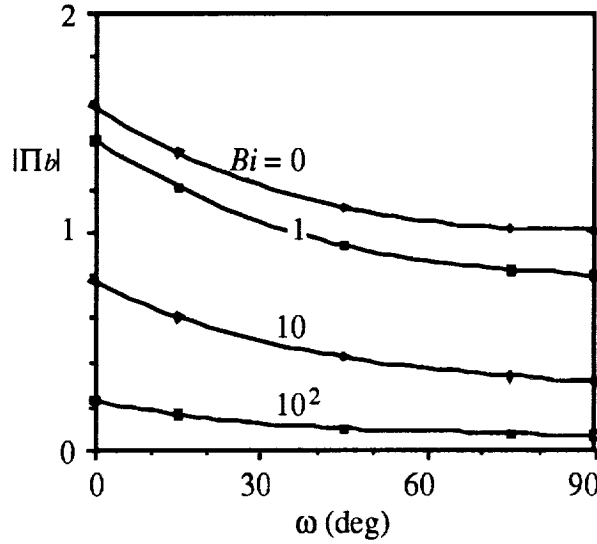
Figure 14. Basic state Π_b versus ω and Bi.

between the liquid and vapor. With $Bi \to 0$, however, the surface assumes the side wall value of temperature and approaches the nonequilibrium limit. As expected, $|\Pi_b|$ tends to increase as $\omega \to 0$, especially at lower Bi, because the temperature becomes more uniform and $|T| \to 1$ along the entire length of the meniscus. As $Bi \to 0$, $|\Pi_b|$ approaches a maximum limit represented by the $Bi = 0$ curve, which corresponds to the surface area as a function of ω.

By employing the boundary conditions for temperature and stress, one can derive additional parameters that characterize the nonlinear influence of Bi and ω on the thermocapillary stress distribution and its resulting traction along the meniscus. Although fluid convection and stress are ignored by definition of the basic state, the surface temperature distributions yield a reference $\partial T/\partial s$ from which a normalized thermocapillary stress, based only on the nonlinear influence of Bi and ω, can be calculated. This derived parameter can then be used later to assess the influence of convection on the actual force applied on the surface.

One begins by expressing the temperature gradient $T_{,i}$ as the product of a magnitude and a unit vector g_i parallel to $T_{,i}$, that is:

$$T_{,i} = \frac{\partial T}{\partial x_i} = |T_{,(i)}| g_i ,$$ (119)

where

$$|T_{,(i)}| = \sqrt{T_{,j} T_{,j}} .$$

One then substitutes equation (119) into the tangential momentum balance and heat transfer conditions (equations (150) and (154), respectively), combines the two equations by eliminating the common $|T_{,(j)}|$ term, and obtains an expression for the magnitude of the local thermocapillary stress in terms of Bi and Re.

$$\tau_{ij} n_j s_i = -ReBiT \frac{g_k s_k}{g_j n_j} .$$ (120)

In deriving this expression, one assumes a nonzero heat flux between the liquid and vapor (i.e., $Bi \neq 0$). BiT in equation (120) represents the local heat transfer rate normal to the interface, while $g_k s_k / g_j n_j$ is the ratio between the tangential and normal components of interfacial heat flux. Equation (120) conveniently separates the linear contributions of Re and Bi from the nonlinear effects of Bi and ω. Since equation (120) acts in the direction of the tangent vector s_m, the components of the thermocapillary stress can be expressed as:

$$\tau_{ij} n_j s_i s_m = ReBi\theta_m \ , \tag{121}$$

where

$$\theta_m = -T \frac{g_k s_k}{g_j n_j} s_m \ .$$

The vector θ_m is termed the modified stress and embodies the nonlinear influence of Bi and ω on the actual stress. $Bi\theta_m$ in equation (121) is the m-component of the temperature gradient parallel to the surface. Since Re represents the ratio of thermocapillary stress to tangential heat flux, $Bi\theta_m$ is the portion of the surface heat flow contributing to tangential stress. An expression is obtained for the overall force acting on the surface in the x_m-direction by integrating equation (121) along the upper boundary.

$$F_m = ReBi\Theta_m \ , \tag{122}$$

where

$$\Theta_m = \int_\Gamma \theta_m d\Gamma \ .$$

Θ_m is the nonlinear contribution to the total thermocapillary force in the x_m-direction. In analyzing the magnitude of force variation, one recognizes that $\Theta_m = \Theta_m(Bi,\omega)$, and that Θ_m tends to vary inversely with Bi. Therefore, to obtain an adequate view of the influence of Bi and ω on F_m, F_m is characterized using the product $Bi\Theta_m$ which is termed the Re-normalized thermocapillary force. θ_m and Θ_m are solely functions of surface geometry and the pore temperature field, and reflect the direction of heat transfer relative to the surface.

With heat flow normal to the surface (i.e., $g_k s_k = 0$ and $g_k n_k = 1$), the thermocapillary stress is zero. Alternatively, with a completely insulated boundary ($g_k n_k = 0$), all heat at the surface flows tangentially thus yielding pure thermocapillary flow. Note that equations (121) and (122) cannot be used in this case because the definitions of θ_m and Θ_m were derived assuming the presence of a normal component of heat flux (i.e., $T_{,k} n_k \neq 0$ and $g_k n_k \neq 0$). In the case of a perfectly insulated boundary or state of maximum nonequilibrium (i.e., $Bi = 0$), one resorts to the original expression for thermocapillary stress.

$$F_i = -Re \int_\Gamma T_{,j} s_j s_i d\Gamma \ . \tag{123}$$

θ_i and Θ_i are useful for showing how contact angle and Bi alter the stress distribution along the surface. Figure 15 shows the x_1 and x_2 components of θ_i (i.e., θ_1 and θ_2) plotted as a function of contour position for the superheat condition, and $Bi = 1$, 10, and 10^2. The distributions are shown for the left half of the meniscus. The positive values of θ_1 for $s-s(x_1 = 0.5) < 0$ produce a net force acting on the left half of the meniscus in the x_1-direction. The θ_1 distribution on the right side of the cavity, which is a mirror reflection of the left half about the s-axis and center-line, results in an equal

opposing force that cancels out the contribution from the left side. θ_2, however, is negative and symmetric about the center-line, and yields a net force in the $-x_2$-direction.
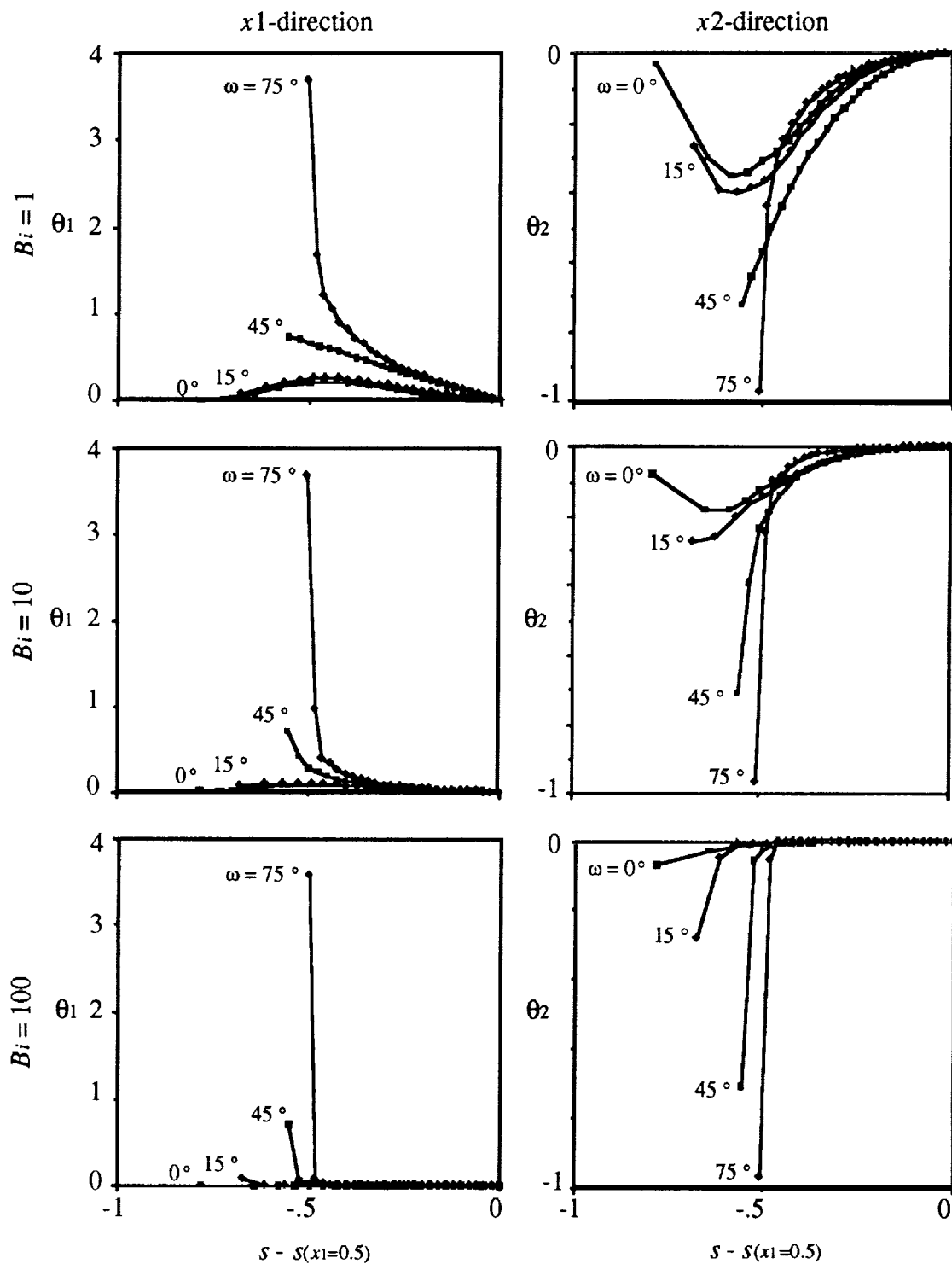
Figure 15. Modified stress versus contour position and ω.

The sensitivity of $\theta_i(s)$ to ω is due to the manner in which the relative orientation between the temperature gradient and surface varies with ω. At the lower limit of Bi ($= 1$), the convection-limited nature of cavity heat transfer yields a surface temperature gradient nearly parallel to the surface, $g_k s_k \rightarrow 1$. Thus, heat flow in the surface direction is greater resulting in a more pronounced modified stress distribution as $Bi \rightarrow 0$.

With low Bi and contact angles, the functional relationship between θ_1 and s exhibits a maximum between the center-line and side wall. At $\omega = 0$, θ_1 vanishes completely in the corners. Not only does the vector contribution in the x_1-direction vanish as $\omega \rightarrow 0$, but the surface becomes parallel to the side wall and orthogonal to the direction of heat transfer, $g_k n_k \rightarrow 1$. The surface in this region is steeper than the temperature gradient, and $g_k s_k$ increases for a certain distance away from the side wall. That is, $\partial\theta_1/\partial s > 0$, and θ_1 increases towards the center of the meniscus. This trend reverses at a certain point since $g_k n_k = 1$ at the center-line. The reversal point at which $g_k n_k$ begins to increase depends on Bi and ω.

With increasing ω, the point of maximum θ_1 shifts to the contact line until $\omega \approx 45°$ where the maximum coincides with the side wall. Above this value of ω, the maximum disappears and $g_k s_k$ decreases continuously from the side wall to the center of the cavity. As $\omega \rightarrow 90°$, the location of maximum $g_k s_k$ remains at the side wall, and θ_1 increases relative to the rest of the cavity.

θ_2 exhibits the same sensitivity to $g_k s_k$ and T. Here, however, the vector component of θ_2 (i.e., s_2) near the side walls increases as $\omega \rightarrow 0$. Unlike θ_1, this tends to offset the decrease in the magnitude of modified stress that occurs with reduced contact angles. Most importantly, it can drastically reduce and eliminate the contribution of x_2-component stress as $\omega \rightarrow 90°$. This unique behavior is illustrated in figure 15 which shows that $|\theta_2|$ can be higher for low ω in the middle of the cavity. Another interesting aspect of θ_2 pertains to the case of a flat surface. Taking the limit of θ_2 as $\omega \rightarrow 90°$, in which s_1 and $g_2 \rightarrow 1$, one obtains $\theta_2 \rightarrow s_2/g_2$ ($= -1$). One can see that g_2 is nonzero at a point immediately adjacent to the side wall, and for $x_1 \neq 0$ or 1, $\theta_2 = 0$. This is the limiting case of the x_2-component stress for a flat surface.

Bi has a strong influence on θ_i as shown by comparing θ_1 and θ_2 for $Bi = 1$ and 10. At larger values of Bi, the temperature gradient becomes increasingly normal to the surface, and the contribution of $g_k s_k$ between the side wall and center-line is reduced. The lowering of θ_1 and θ_2 about the center-line is caused by the decrease in T and more perpendicular orientation of $T_{,i}$ with respect to the surface. From a thermal standpoint, the interfacial temperature distribution is less conducive to thermocapillary stress because of the increased normal component of heat flux. This trend reflects a shift of the temperature distribution towards the side wall, which serves to decrease θ_i but increase the actual stress $Bi\theta_i$ due to the higher overall gradient.

With $Bi = 10$ and $\omega = 45°$, the variation of θ_1, which decreased linearly with s at $Bi = 1$, follows the same general form as θ_1 with $\omega = 75°$. In addition, the increasingly normal temperature gradient is noted by the disappearance of θ_i extrema at the smaller values of ω. θ_2 follows the same trends as θ_1, but because of the higher s_2 near the interline, the x_2-contribution to stress is higher for low contact angles. The value of θ_2 for all contact angles decreases with Bi due to the higher temperature gradients and stress in the side wall regions. At the upper limit of Bi ($= 10^2$), the stress force concentrates even more in the side wall regions for both the x_1 and x_2 components. At low

contact angles, however, the x_1 component vanishes almost entirely (compared to a flat interface), while in the x_2-direction, it remains relatively high.

An important aspect of this problem is the manner in which $|Bi\Theta_i|$ varies as a function of ω. As the meniscus flattens ($\omega \to 90°$), the magnitude of Θ_1 exerted on each half-cavity surface $\Theta_1{}^h$ tends to increase. Because of mirror symmetry, however, the total Θ_1 contribution equals zero. The only component remaining when one views the stress force in its entirety is Θ_2. The relationship between $|Bi\Theta_2|$ for the basic state (i.e., $|Bi\Theta_{2b}|$) and ω is shown in figure 16 At $Bi = 1$, $|Bi\Theta_{2b}|$ is equivalent to $|\Theta_{2b}|$, and reducing ω generally increases the total applied force in either the $-x_2$ or $+x_2$-direction. $|Bi\Theta_{2b}|$ actually reaches a maximum at $15° < \omega < 45°$ and decreases slightly as $\omega \to 0$. The important point is that the total force disappears with a flat surface.
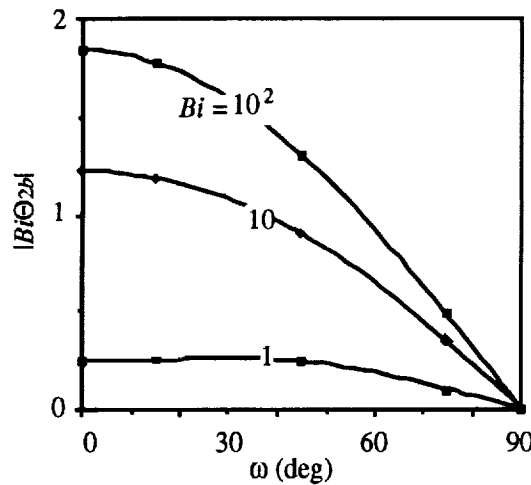
Figure 16. Basic state $Bi\Theta_{2b}$ versus ω and Bi.

Although the nonlinear contribution to the total force decreases as $Bi \to \infty$, the linear influence of Bi offsets this and yields an increase in $|Bi\Theta_2|$. The sensitivity of $|Bi\Theta_2|$ to contact angle tends to increase as $Bi \to \infty$ due to the concentration of temperature change in the interline region. As $Bi \to \infty$, $\partial T/\partial s$ and a greater portion of the total stress become restricted to regions in which $|s_2|$ is high. Hence, with lower ω, more of the total stress force is directed in the x_2-direction.

C. Pure Evaporation and Condensation

Although both modes of interfacial mass transport are based on the same kinematic and equilibrium assumptions for the liquid/vapor interface, their influence on the steady-state temperature field, thermal potential, and modified stress force is quite different. To model pure evaporation and condensation, the thermocapillarity is ignored by setting $Ma = 0$. This minimizes the tangential component of surface velocity and causes formation of a vertically-oriented flowfield. The strength and structure of the field, particularly near the surface, depend on Rs and Bi. An example of the stream function and temperature distribution for steady-state condensation and evaporation with $Rs = 10^{-1}$ and $Bi = 1$ is shown in figure 17. Note that the reference value of Ψ is set to 0 at the base of the center-line. The Ψ values for the two cases are opposite in sign to reflect the different directions of flow relative to the meniscus.
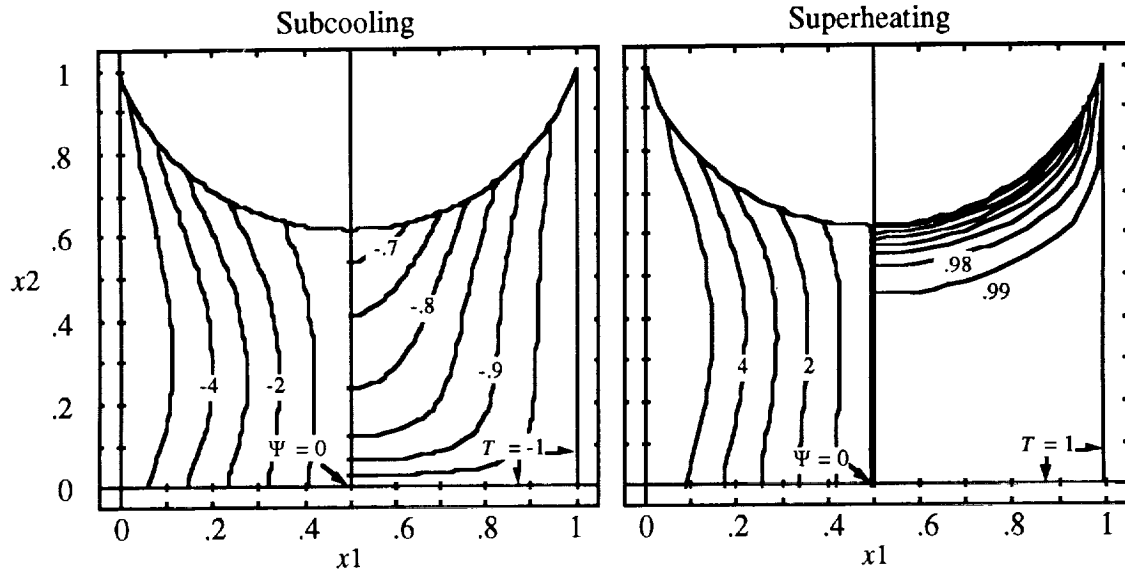
Figure 17. Pure interfacial flow ($Rs = 10^{-1}$ and $Bi = 1$).

Although no surface traction is present, a tangential component of surface velocity does exist due to curvature and the nonuniformity in temperature. This phenomenon is illustrated in figure 17 by the slight bending of the streamlines toward the side wall near the meniscus. The directional bias arises because the maximum temperature difference and driving potential for mass transport occurs at the contact line. The main consequence of this phenomenon is an increase in local vorticity near the surface and strengthening of circulation on either side of the cavity.

The circulation in each half-cavity, which is defined as $C = \int_\Gamma V_i s_i d\Gamma$, arises naturally from the angular momentum caused by inequality between velocities along the center-line and no-slip region of the side wall. For the left half-cavity, the equation for C is integrated counterclockwise along the composite surface Γ containing the vertices $(x_1,x_2) = (0,1)$, $(0,0)$, $(0.5,0)$, and $(0.5,y_0^{(s)})$. As will be discussed later, half-cavity circulation appears to be an important aspect of the flow structure, and is useful for assessing the competition between simultaneously occurring convection modes.

Comparing the cases in figure 17 points to several important differences between the two modes of mass transfer. One is the generally lower flow intensity of condensation compared to evaporation. With evaporation, the magnitude of Π increases from a basic state value of 1.17 to 1.27 which equates to raising T_{avg} from 0.86 to 0.94. The higher value of $|\Pi|$ arises from the compression of isotherms caused by convection of superheated liquid from the bottom of the cavity, which tends to reinforce itself by raising the temperature and velocity normal to the surface. With condensation, the downward direction of flow distends the temperature field about the surface and compresses the isotherms towards the bottom of the cavity. This causes a lowering of $|\Pi|$ from 1.17 for the basic state to 1.13 with convection, an increase in T_{avg} from –0.86 to –0.83 and a reduction of Q into the cavity. The condensation of warm liquid on the surface raises temperature, thus suppressing the driving potential for mass flow into the cavity.

Another important difference is the magnitude of half-cavity circulation. For the subcooling and superheating cases in figure 17, $C = -17.71$ and 19.49, respectively. The larger circulation

magnitude for evaporation is primarily due to the flowfield's higher mass flux and velocity along the cavity center-line. This reflects the increased $|\Pi|$ caused by compression of isotherms near the surface. Figure 17 also indicates that the surface velocity contribution as a fraction of total circulation is greater for condensation as evidenced by the more pronounced side wall bias of Ψ near the meniscus. At low Bi, however, the surface velocity contribution is small compared to center-line velocity.

The effect of increasing Bi to 10, while holding $Rs = 10^{-1}$, is shown in figure 18. At larger values of Bi, the temperature distribution approaches the vapor value in the middle of the cavity and acquires a larger gradient near the side walls. This change in heating characteristics reduces flow intensity and yields less isotherm deformation, as evidenced by the lowering of $|\Pi|$ for subcooling and superheating to 0.56 and 0.70, respectively. The associated reduction in center-line flow translates to a decrease in circulation magnitude (i.e., $C = -11.46$ and 13.19 for subcooling and superheating, respectively). Although center-line flow is reduced, the more pronounced Ψ-bias indicates a growing contribution to circulation from surface velocity.
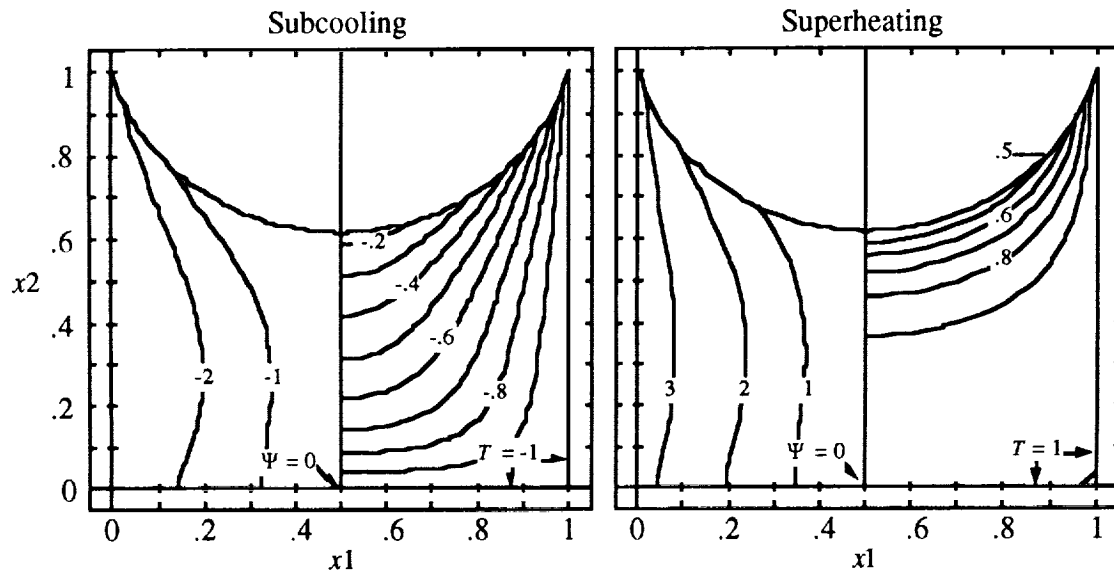
Figure 18. Pure interfacial flow ($Rs = 10^{-1}$ and $Bi = 10$).

Increasing Bi reduces the influence of convection on the pore temperature distribution and heat transfer. This is especially apparent in figure 19 which shows the Ψ and T distributions for $Bi = 10^2$. The Ψ profiles are essentially the same for evaporation and condensation, and the isotherms are very similar to those for the basic state. The circulation magnitudes also decrease to nearly equivalent values ($C = -4.43$ and 4.76 for subcooling and superheating, respectively) due to the drop in center-line flow and diminishing influence of convection on the temperature field.

The Bi variation illustrated in figures 17 to 19 actually reflects a lowering of the evaporation number E while holding Rs constant. Care must be taken when varying Bi in cases involving condensation or evaporation, because $Bi = Bi(E,Rs)$. Proper assessment of the influence of Rs, which arises from application of equilibrium assumptions to the kinematic condition for vapor/liquid mass exchange, must account for the contribution of Rs to Bi. This can be done by comparing figure 18 with

58
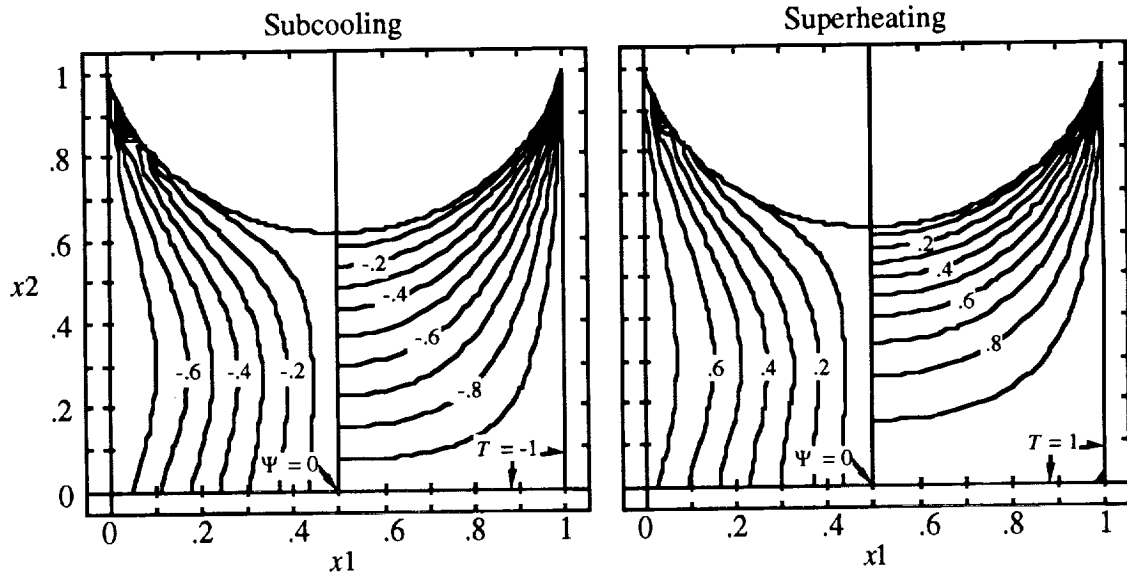
Figure 19. Pure interfacial flow ($Rs = 10^{-1}$ and $Bi = 10^2$).

Fig. 20 which illustrates the effect of reducing Rs from 10^{-1} to 10^{-2} while holding $E = 10$. At the lower value of Rs, the interfacial flux and flowfield strength for evaporation increase and cause gr er isotherm compression near the surface. Although the interfacial temperature gradient is substantially larger, the total degree of nonequilibrium represented by Π and T_{avg} decreases from 1.27 to 1.21 and 0.94 to 0.89, respectively. This is consistent with the definition of Rs from section III in that as $Rs \to 0$, $Bi \to \infty$ and $T \to 0$ (or $\Pi \to 0$). Although the decrease in Rs results in greater equilibrium between the vapor and liquid, the associated convection and deformation of the temperature field increases the T gradient near the surface and degree of nonequilibrium in the liquid.
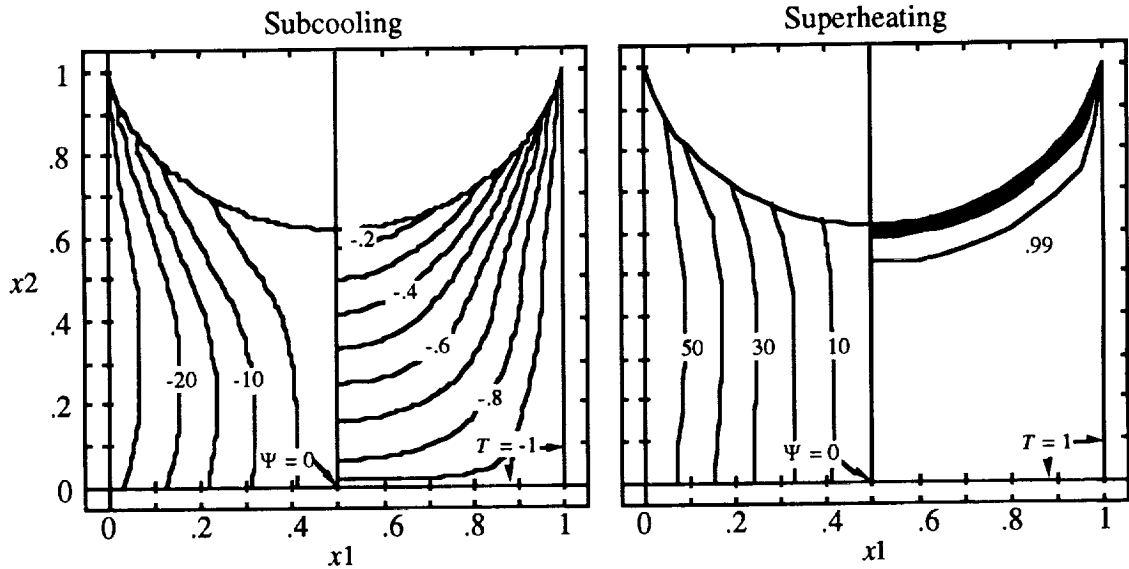
Figure 20. Pure interfacial flow ($Rs = 10^{-2}$ and $Bi = 10$).

With condensation, reduction of Rs from 10^{-1} to 10^{-2} also increases flowrate. However, the Ψ-bias and temperature gradient in the corners becomes greater due to an increase in interfacial temperature. Unlike evaporation, the downward flow of condensing liquid promotes equilibrium along the interface by compressing the isotherms towards the bottom of the cavity. The reduction in the magnitudes of $|\Pi|$ (from 1.13 to 0.53) and T_{avg} (from 0.83 to 0.39) is much greater than with evaporation and indicates a higher level of equilibrium along the interface. Because of the lower $|\Pi|$, the flow intensity for condensation is less than evaporation. In addition, there is an increase in temperature gradient and nonequilibrium along the other boundaries.

The influence of both mass transfer modes on interfacial transport, circulation, and, in the case of a nonzero $\partial\gamma/\partial T$, stress is characterized by the departure of temperature from the basic state. Surface temperature distributions for evaporation and condensation with $Bi = 1$, 10 and 10^2 are shown in figures 21 and 22, respectively. Note that these profiles have been superimposed on those for the basic state ($Rs = \infty$) to aid in visualizing the deviation.
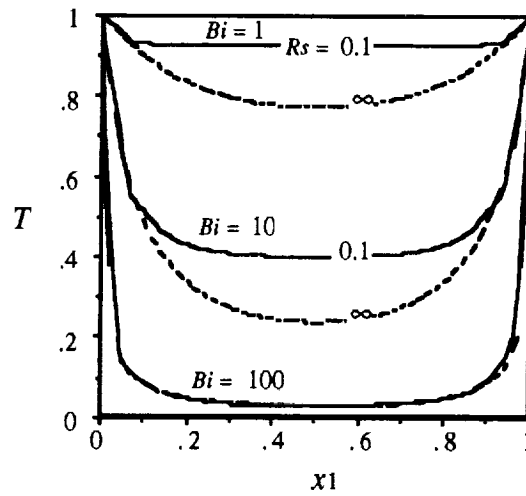
Figure 21. Interfacial temperature (pure evaporation).
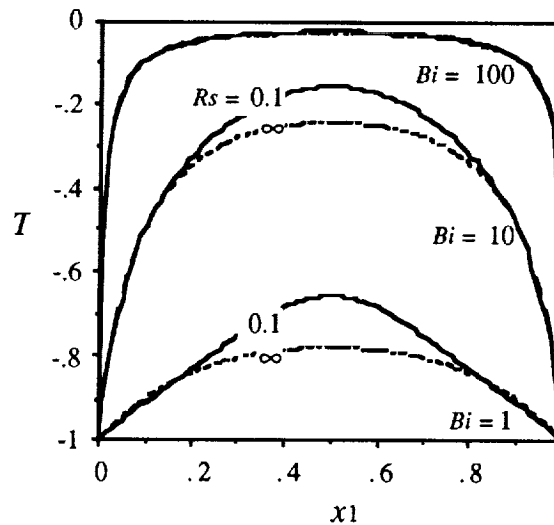
Figure 22. Interfacial temperature (pure condensation).

Figures 21 and 22 confirm the behavior noted previously, namely that evaporation causes a flattening of the temperature distribution and an increase in Π. This is attributed to the replenishment of superheated liquid from the lower boundary which, upon approaching the surface, is directed toward the corners and regions of highest mass/heat transfer. Because the liquid/vapor temperature difference is lower in the center, this hot liquid sweeps along the surface to the sidewalls, thus flattening the temperature distribution and reducing the modified stress force.

With condensation, the surface is heated by the vapor and assumes a higher temperature relative to the basic state. Since the mass flux is concentrated in the contact region, flow is directed into the center of the cavity, which causes a peaking in the temperature profile. Unlike evaporation, this temperature rise reduces the temperature difference between the vapor and liquid and decreases the magnitude of Π. However, the average temperature gradient along the surface is higher than the basic state and yields a larger modified stress force.

The influence of both mass transfer modes on interfacial transport, stress, and, indirectly, circulation is characterized by the departure of Π and Θ_2 from the basic state. With interfacial mass transfer, Π exhibits the same functional dependence as Π_b in that $|\Pi|$ tends to increase with reduced Bi and ω. Deviation from this reference is best expressed using the ratio Π/Π_b. Figure 23 shows Π/Π_b versus Bi and ω at $Rs = 10^{-1}$. The deviation for evaporation is always positive, since $\Pi/\Pi_b > 1$ over the range of interest. With condensation, $\Pi/\Pi_b < 1$, thus indicating a reduction in thermal potential and suppression of interfacial transport. The point of maximum deviation (i.e., extrema of Π/Π_b) represents the transition between convection and conduction-limited heat transfer processes, and occurs between $1 < Bi < 10$. In this range, fluid convection plays a principal role in dictating the temperature distribution. As $Bi \rightarrow \infty$, the deviation from the basic state vanishes (i.e., $\Pi/\Pi_b \rightarrow 1$). The surface and domain temperature distributions become conduction-limited, and are affected less by fluid motion within the cavity. With vanishing Bi, however, the more uniform liquid temperature restricts the magnitude of temperature variation near the surface, and Π/Π_b approaches unity.
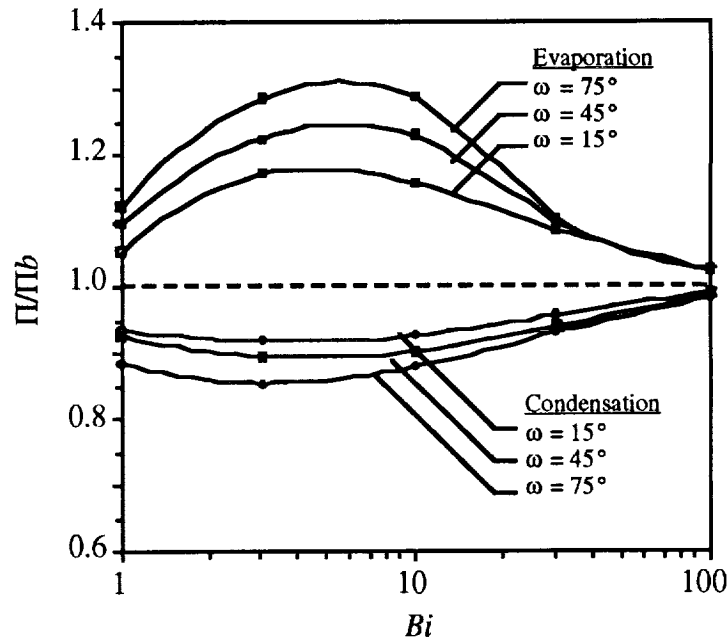
Figure 23. Π/Π_b versus Bi and ω (interfacial flow).

Evaporation and condensation also have different effects on the temperature gradient and modified stress along the surface. Figure 24 shows Θ_2/Θ_{2b} as a function of Bi and ω. The fact that $\Theta_2/\Theta_{2b} < 1$ for evaporation confirms that this mode of interfacial transport tends to reduce the average temperature gradient and yields a lower total stress force in the x_2-direction than the basic state. It also suggests that evaporation should suppress thermocapillarity when both convection modes are present. At the upper limit of Bi ($= 10^2$), there is little difference in the stress distributions ($\Theta_2/\Theta_{2b} \cong 1$) since the temperature deviation is so small. With lower Bi, most of the change in temperature relative to the basic state occurs in the center of the pore, but the temperature distribution and gradient near the side wall are essentially the same. Hence, the force ratio remains constant. With further reductions in Bi, the maximum change in interfacial temperature gradient shifts toward the side walls and at a certain point begins to reduce the gradient. This reduction continues to a minimum at $Bi \cong 1$. Further reductions in Bi, while not shown on figure 24, must cause an increase in Θ_2/Θ_{2b}, because as $Bi \to 0$, the basic and convective state temperatures approach unity, and $\partial T/\partial s \to 0$ for both cases.
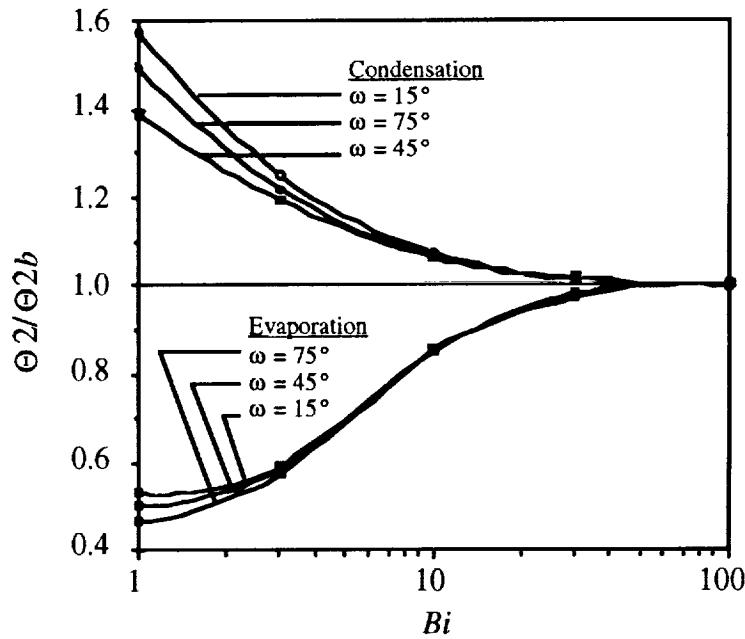
Figure 24. Θ_2/Θ_{2b} versus Bi and ω (interfacial flow).

Condensation exhibits an opposite effect on the thermocapillary force. That is, $\Theta_2/\Theta_{2b} > 1$ and the stress increases with lower Bi. As $Bi \to \infty$, the behavior is very similar to evaporation and $\Theta_2/\Theta_{2b} \to 1$. As Bi is reduced, however, the accumulation of heated liquid in the center sustains a higher gradient along the surface. The gradient near the side wall for the basic state decreases faster than that for the convective regime which assumes a nearly constant distribution from the side wall to the center of the cavity. Note that with convection, the change in the gradient becomes increasingly isolated in the center of the meniscus. Therefore, the force ratio, which is mostly dictated by the gradient in the side wall regions, is greater and continues to increase as $Bi \to 0$. Figure 24 suggests that Θ_2/Θ_{2b} increases without limit with $Bi < 1$. This, however, cannot be true since $\Theta_2 \to 0$ as $Bi \to 0$. It is likely that, as with evaporation, $\Theta_2/\Theta_{2b} \to 1$ as $Bi \to 0$.

The influence of condensation and evaporation on circulation is summarized in figure 25. The most apparent trend is the higher circulation associated with small contact angles. This is due to the larger center-line velocities resulting from the increase in $|\Pi|$ and exposed surface area. With large ω, the streamlines become more perpendicular to the surface, which further reduces the contribution of surface velocity to circulation intensity. Another trend is the higher circulation associated with evaporation. This is especially evident at low Bi due to evaporation's higher $|\Pi|$ and center-line velocity. At the upper Bi limit, however, the circulation for condensation appears to be greater at large contact angles (i.e., $\omega = 45°$ and $75°$). In this regime, the center-line velocity is low compared to the surface component. Hence, the larger surface temperature gradients associated with condensation yield higher circulation intensities than evaporation.

The differences in $|C|$ for condensation and evaporation appear to vanish at the lower and upper Bi limits due to the equivalence of surface temperatures. As $Bi \to 0$, the surface temperature approaches unity and yields a normal flux that predominates over a vanishing tangential component. The center-line velocity approaches a value that is entirely a function of ω and Rs, and is equivalent in magnitude for both modes. Therefore, the magnitudes of $|C|$ should converge to the same value as $Bi \to 0$. With larger values of Bi, the center-line contribution decreases but is partially offset by the increase in surface velocity. The net effect is a decrease in circulation due to a reduction in throughput. As $Bi \to \infty$, the contribution from the center-line vanishes and the circulations for both cases approach zero.
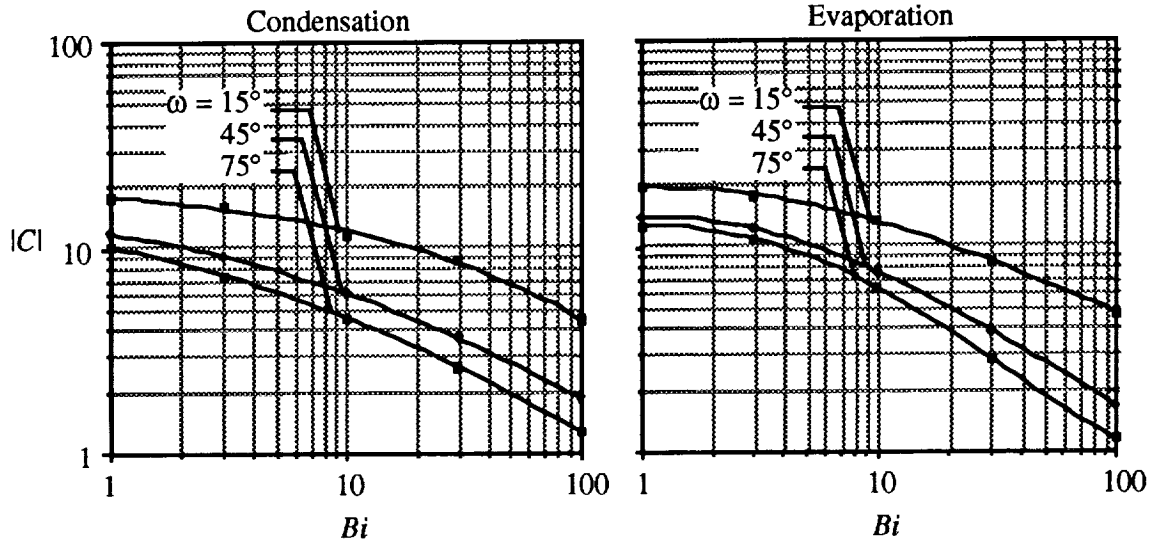
Figure 25. Half-cavity circulation versus Bi and ω (interfacial flow).

D. Pure Thermocapillary Flow

To examine pure thermocapillary flow, the interfacial resistance is fixed at a high value ($Rs = 10^3$) to eliminate the normal component of velocity and ensure parallel flow at the surface. The influence of Ma and Bi is first investigated by considering the values $Ma = 10^2$ and 10^3, and $Bi = 1$, 10, and 10^2 at a fixed ω of $15°$. Figure 26 shows the steady-state Ψ and T distributions for a subcooled and superheated cavity with $Bi = 1$ and $Ma = 10^2$. For both heating modes, the thermocapillary stress, which acts opposite to the surface temperature gradient, establishes twin counter-rotating vortices on either side of the cavity. With subcooling, the temperature gradient is

negative in the x'-direction, and causes a traction toward the corners. This establishes counterclockwise and clockwise rotations in the left and right sides of the cavity, respectively. With superheated boundaries, $\partial T/\partial x' > 0$, and the traction on either side is directed into the center or $-x'$-direction. The sense of cell rotation for subcooling and superheating are opposite, as indicated by the positive and negative values of streamfunction. The close similarity between the isotherms in figure 26 and the corresponding basic state in figure 11 indicates that at low Ma the flow is too weak to cause appreciable deformation of the temperature field. Conduction remains the primary mode of heat transfer.
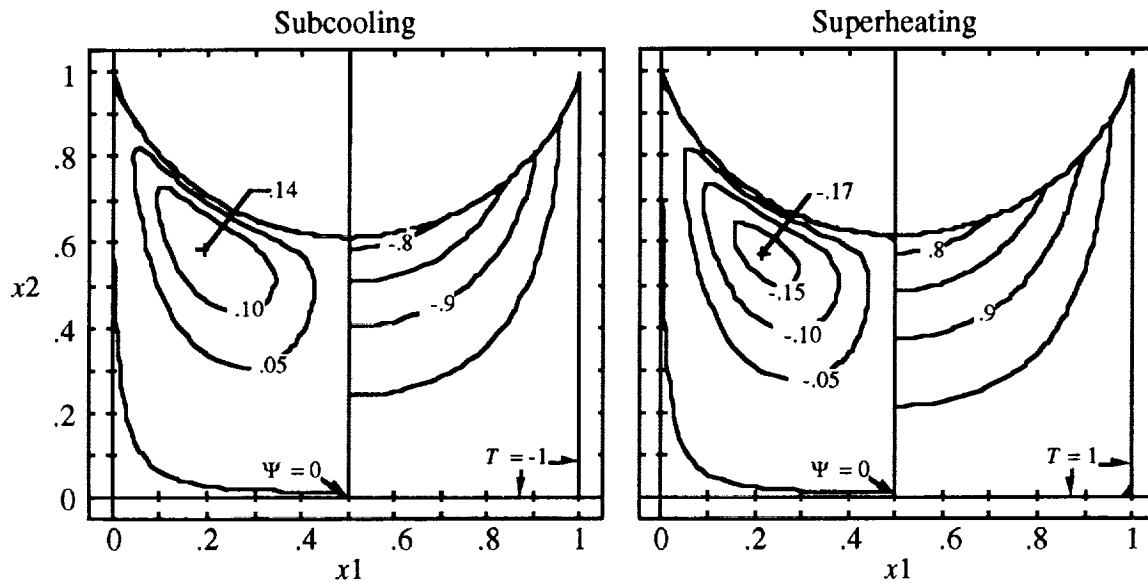
Figure 26. Pure thermocapillary flow ($Ma = 10^2$ and $Bi = 1$).

At higher Ma, however, convection parallel to the surface causes a significant deviation from the basic state temperature. This is particularly evident in figure 27 which shows the Ψ and T distributions for $Ma = 10^3$. Apart from a nearly order of magnitude increase in cell strength, reflected by the larger values of Ψ, increasing the sensitivity of surface stress to temperature substantially alters the surface and domain temperature distributions from the basic state reference. In the case of subcooling, surface fluid convected towards the sides of the cavity deforms the isotherms away from the center-line and establishes a higher $|dT/dx'|$ in the vicinity of the side wall. Return circulation from the lower portion of each cell convects subcooled liquid upwards through the center, thus compressing the isotherms and causing a temperature depression in the middle of the surface. With superheating, the isotherms near the surface deform into the center of the cavity due to the transport of heated liquid from the side walls. This circulation, which is consistently stronger than subcooling, causes isotherm deformation in the direction of flow and a reduction of $|dT/dx'|$ near the corners. The isotherms about the axis of symmetry extend down into the fluid due to the transport of cool liquid from the surface.
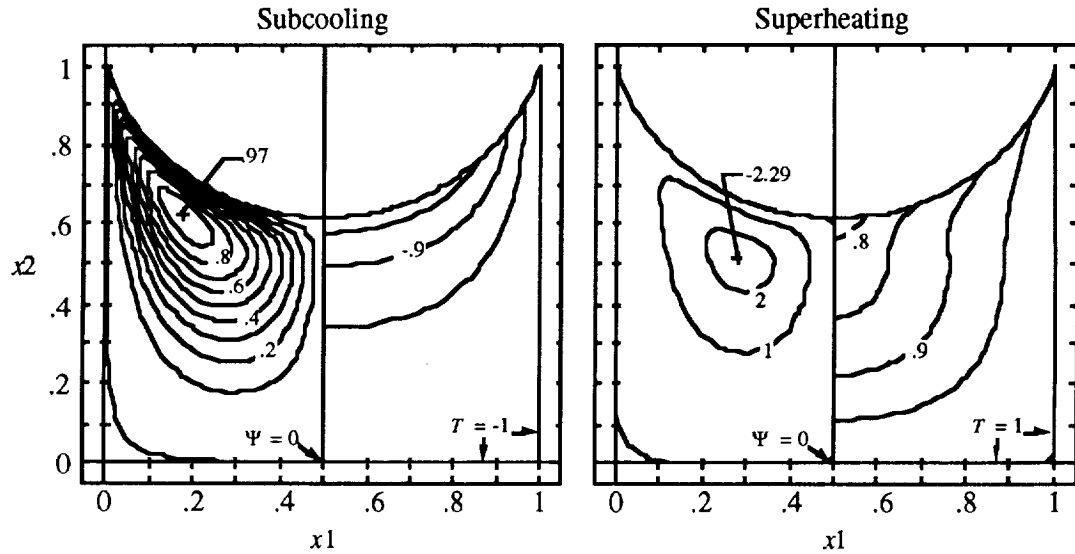
Figure 27. Pure thermocapillary flow ($Ma = 10^3$ and $Bi = 1$).

The influence of raising Bi is shown in figures 28 and 29 which depict Ψ and T distributions for $Ma = 10^2$ and 10^3 at $Bi = 10$. An obvious difference between these cases and those for $Bi = 1$ is the marked increase in circulation intensity arising from increased temperature variation and stress. At $Ma = 10^2$, deformation of the temperature field is more extensive than figure 26 ($Bi = 1$). However, it is relatively small compared to the $Ma = 10^3$ case, where one encounters considerably more isotherm deformation than figure 27. With subcooling and $Ma = 10^3$, the flow assumes the same structure as $Bi = 1$, but the transport of heated liquid in the x'-direction shifts the maximum temperature to a point midway between the center-line and side wall. With superheating, the circulation cell shifts toward the cavity center and causes formation of a temperature depression in the middle of the pore. The gradient associated with this depression results in a centrally located stress concentration directed towards the center-line.
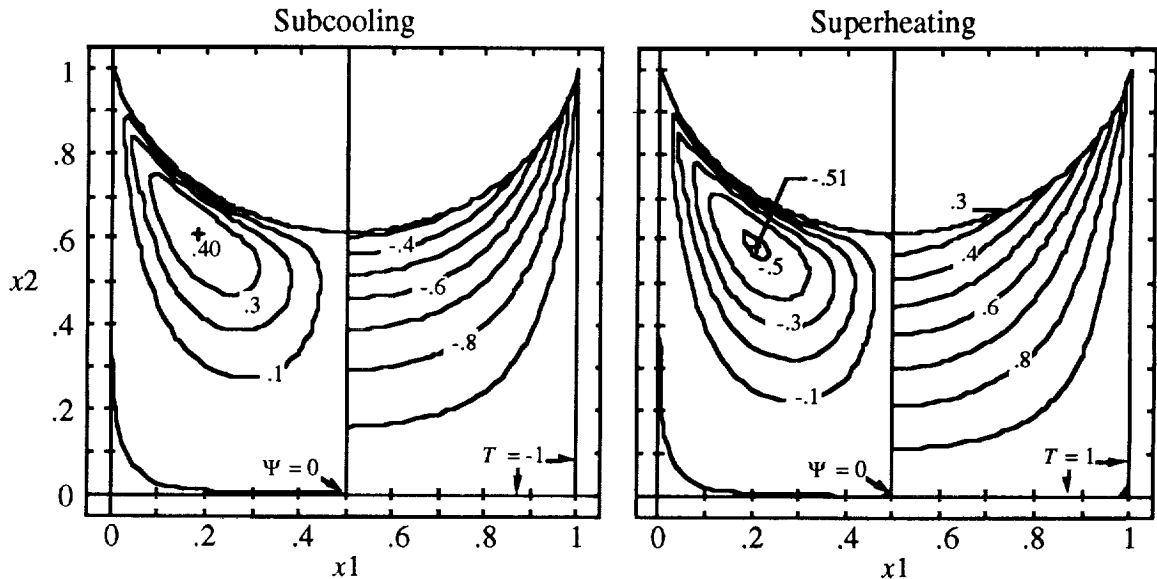
Figure 28. Pure thermocapillary flow ($Ma = 10^2$ and $Bi = 10$).
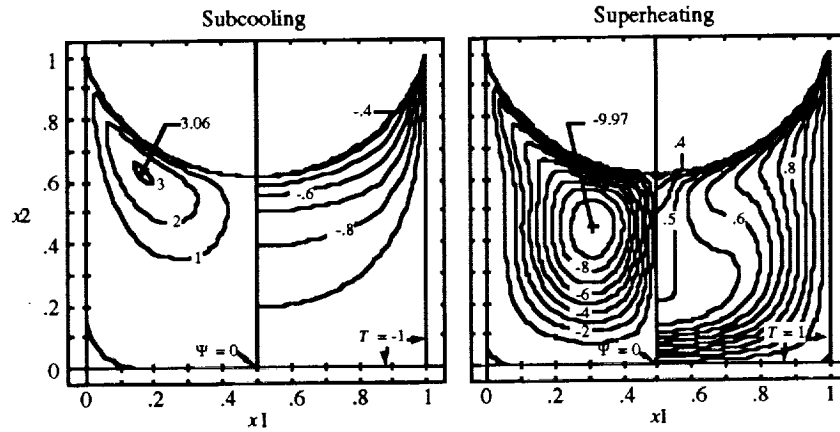
Figure 29. Pure thermocapillary flow ($Ma = 10^3$ and $Bi = 10$).

The same trends apply at the upper limit of Bi (= 10^2), shown in figures 30 and 31. Although the flowfields are stronger than the previous cases, at low Ma (= 10^2), the difference in isotherm distributions for the two heating modes vanishes as $Bi \to \infty$.
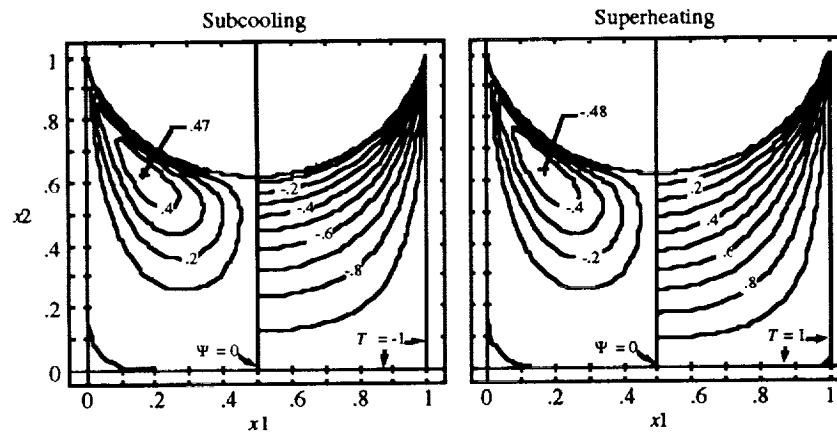
Figure 30. Pure thermocapillary flow ($Ma = 10^2$ and $Bi = 10^2$).
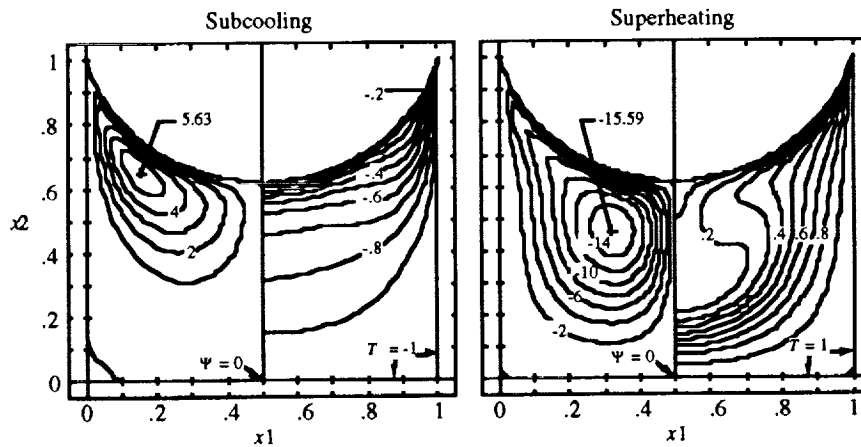
Figure 31. Pure thermocapillary flow ($Ma = 10^3$ and $Bi = 10^2$).

All the cases investigated so far indicate that superheating consistently yields a stronger circulation than subcooling. This difference is primarily due to the manner in which the circulation cells situate near regions of concentrated stress. That is, the cells shift towards the side wall with subcooling, but move towards the center-line when superheated. The viscous losses associated with these positions are quite different. At the side wall, subcooled flow accelerated along the surface encounters the stationary boundary at a higher relative velocity than the superheated regime, which meets it at the bottom of the cell. Thus, the viscous losses along the side wall are greater for the subcooled regime. Along the center-line, however, both modes experience a slip condition caused by opposing flow from the other side of the cavity. Although the viscous losses here are greater for superheating, the losses for both modes are less than those at the side wall. Consequently, the superheated flow regime incurs less viscous losses and a higher circulation than subcooling.

Figures 32 and 33 depict the interfacial temperature profiles for subcooling and superheating with $Ma = 10^3$ and $Bi = 1$, 10 and 10^2.
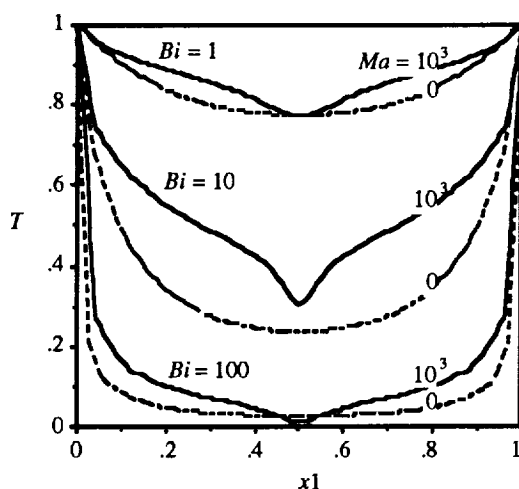
Figure 32. Interfacial temperature (superheating).
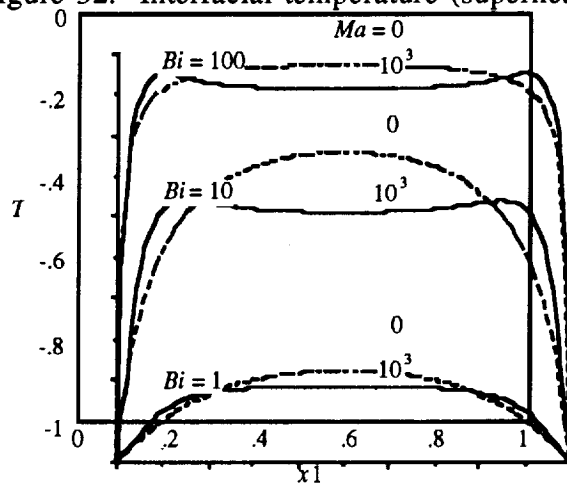
Figure 33. Interfacial temperature (subcooling).

In both cases, the deviation from the basic state distribution is caused by the accumulation and convection of heated liquid along the surface. With subcooling, the liquid is convected towards the side wall thus shifting the temperature distribution and isotherms to either side of the cavity. There is also a competing upward circulation of cooler fluid along the center-line which reduces surface temperature and causes a localized depression in the temperature profile. This competition arises because upwelling opposes the surface temperature gradient and can cause $\partial T/\partial s$ to vary in sign between the center-line and side wall.

With superheating, liquid heated at the side walls flows to the center under the action of thermocapillary stress, while fluid heated at the bottom is circulated and directed at the corners. The direction of convective heat transport coincides with that for surface conduction, and $\partial T/\partial s$ changes sign only at the center of the meniscus. Again, one observes the formation of a temperature depression near the center-line caused by the accumulation and downward circulation of heated liquid.

The response of surface temperature to thermocapillary convection yields a more complex deviation from the basic state thermal potential and modified stress force than pure evaporation or condensation. Figure 34 shows Θ_2/Θ_{2b} and Π/Π_b as functions of Bi for the cases in figures 26 to 31.
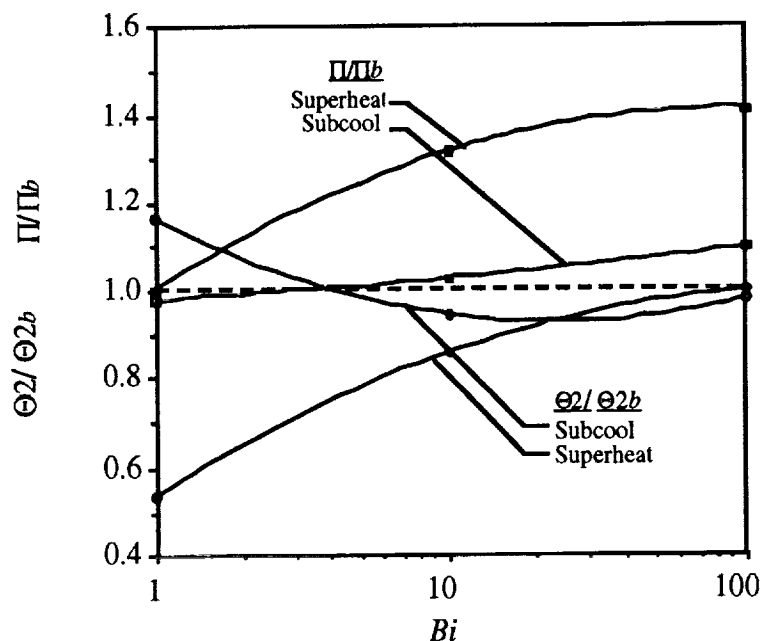
Figure 34. Θ_2/Θ_{2b} and Π/Π_b versus Bi (thermocapillary flow).

One sees that the deviation in Π for both heating modes increases with Bi. With superheating, hot liquid flows from the side walls to the center-line, and from the bottom to the corners. Cellular convection in this case is complementary with the interfacial temperature gradient and tends to raise Π. With subcooling, the heated liquid convected along the surface tends to raise interfacial temperature and lower Π. However, the upward circulation of cooler fluid causes a localized depression in the temperature profile that tends to increase Π. Although the influence of upwelling increases with Bi, Π/Π_b for subcooling is always less than superheating. In fact at low Bi, surface convection dominates the interfacial temperature profile, and $\Pi/\Pi_b < 1$.

The magnitude of Θ_2 for superheating is always less than the basic state because of its lower average surface temperature gradient. One sees that this difference is greatest at the lower limit of Bi, and as $Bi \to \infty$, $\Theta_2/\Theta_{2b} \to 1$. An almost opposite trend occurs with subcooling. At low Bi, interfacial convection shifts the temperature gradient to the side wall, thus increasing the magnitude of $\partial T/\partial s$ and the contribution of s_2 to Θ_2. With higher Bi, however, this effect is increasingly offset by the temperature depression caused by upwelling, which opposes the thermocapillary-induced thermal gradient and stress. The competition between these two effects yields a minimum condition for Θ_2/Θ_{2b} at $Bi \approx 30$. Above this value, upwelling predominates, and Θ_2/Θ_{2b} increases such that $\Theta_2/\Theta_{2b} \to 1$ as $Bi \to \infty$.

Figure 35 illustrates the variation in circulation intensity with Bi for $Ma = 10^2$ and 10^3. As explained before, the circulation for superheating is higher due to the lower viscous losses incurred along the side wall. At low Ma, the values of $|C|$ for both modes converge to a common value due to the reduction in isotherm deformation at low and high Bi.
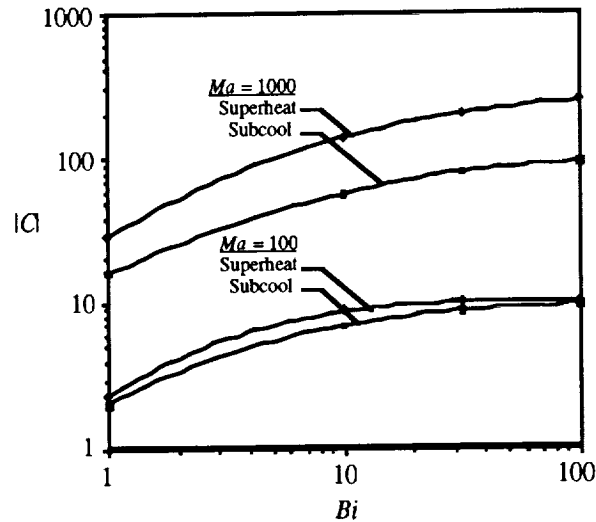
Figure 35. Half-cavity circulation versus Bi (thermocapillary flow).

As $Bi \to 0$, the surface temperature becomes more uniform and less sensitive to convection. Furthermore, the temperature gradient vanishes and $C \to 0$. On the other hand, as $Bi \to \infty$, the gradient becomes concentrated in the side wall region, and the flow becomes conduction-limited. The difference in isotherm deformation and circulation intensity for the two modes vanishes again. At the upper limit of Ma (= 10^3) and low Bi, the circulation exhibits the same trends. At high Bi, however, surface temperature is much more sensitive to superheating and acquires a distribution that increases the thermocapillary stress force relative to subcooling, as shown in figure 34. The divergence between the C-curves in figure 35 shows that this difference in behavior persists as $Bi \to \infty$.

One of the most important consequences of thermocapillary flow with small contact angles is the large dynamic pressure gradient encountered in the contact region. This is illustrated in figure 36 which shows the P_d distributions for $Ma = 10^3$ and $Rs = 10^3$ at a contact angle of $15°$. With subcooling, surface flow towards the side walls causes a rise in pressure at the corners and a suction in the middle of the meniscus. With superheating, however, the flow is reversed and causes a suction at the side walls. In all cases, the pressure distribution in the center of the cavity is relatively constant, and becomes significant only at the sides of the pore.
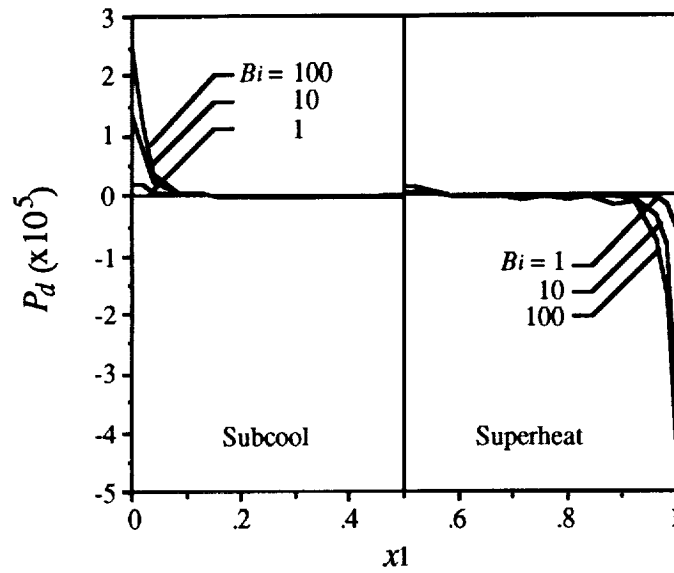
Figure 36. Dynamic pressure versus x_1 ($Ma = 10^3$).

The magnitude of the pressure gradient appears to be a strong function of the thermocapillary stress, as illustrated by the sensitivity of P_d to Bi. Since Ma is held constant, the larger values of Bi correspond to higher stress forces and circulations in the cavity. Because the circulation for superheating is consistently higher than that for subcooling, the variation in pressure magnitudes for the two heating modes is most likely related to the difference in their circulation intensities.

The large pressure magnitudes in the contact region are caused by the accommodation of flow through an area that becomes increasingly constricted with smaller contact angles. The pressure must increase to accommodate the higher viscous stress in the side wall region near the corners. It can be shown using a simple force and momentum balance about the corner that $P_d \ll 0$ for superheating and $P_d \gg 0$ for subcooling. The difference in sign is due to the different directions of thermocapillary stress application. From a physical standpoint, pressure acts as a force applied at the corner that balances changes in fluid momentum, thermocapillary stress and friction. Because the momentum change is very sensitive to restrictions on flow area, the pressure gradient is a strong function of contact angle.

It was shown in section VI.B that contact angle has a notable effect on the modified stress force. Therefore, one expects any convective motion that depends on surface area and orientation to be very sensitive to ω. The influence of ω is particularly important when considering highly wetting fluids because, as shown in figure 36, it appears to lead to extremely large dynamic pressure gradients at the meniscus contact line.

To examine the influence of contact angle, Ma and Bi are held at 10^2 and 10, respectively. The lower value of Ma is selected to limit deviation from the basic state and emphasize the influence of ω. Figure 37 shows the steady state stream function and temperature distributions for thermocapillary flow at $\omega = 75°$, $45°$, and $15°$. The most apparent trends are the suppression of circulation at larger contact angles and the higher flow intensities associated with superheating.

70
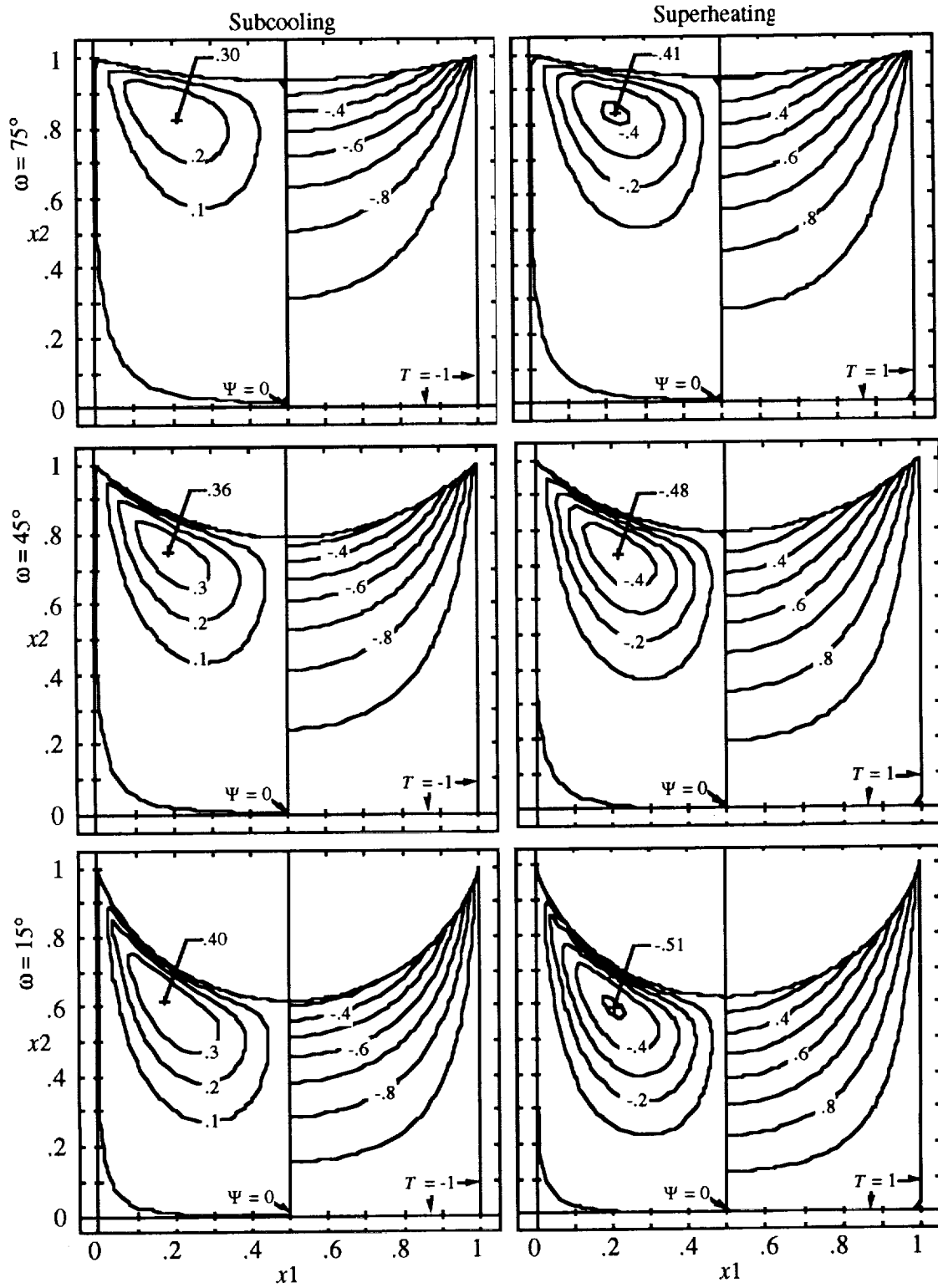
Figure 37. Contact angle sensitivity ($Ma = 10^2$ and $Bi = 10$).

The differences between these six cases are summarized in table 5, which in addition to Π and $Bi\Theta_2$, shows the half-cavity stress force in the x_1-direction $Bi\Theta_1{}^h$, maximum half-cavity

71

stream function $|\Psi|_{max}$, the circulation C, and the difference in interfacial pressure between the contact line and cavity center $|P_d|_{max}$. Since Ma and Bi are the same for all cases, $Bi\Theta_i$ directly reflects the nonlinearity of stress force or Θ_i.

Table 5. Pore characteristics ($Ma = 10^2$, $Rs = 10^3$, and $Bi = 10$).

Mode	ω	$|\Pi|$	$|Bi\Theta_1{}^h|$	$|Bi\Theta_2|$	$|\Psi|_{max}$	C	$|P_d|_{max}$
Subcool	$15°$	0.598	0.388	1.194	0.40	6.81	28,260
	$45°$	0.422	0.594	0.899	0.36	4.59	5,670
	$75°$	0.343	0.760	0.345	0.30	3.06	2,057
Superheat	$15°$	0.603	0.454	1.198	0.51	−7.95	27,480
	$45°$	0.427	0.651	0.896	0.48	−5.69	5,914
	$75°$	0.338	0.801	0.342	0.41	−4.04	1,901

$Bi\Theta_1{}^h$ behaves similarly to the basic state in that it tends to increase with a flatter interface. The total force in the x_2-direction, however, decreases dramatically which approximately coincides with the trend for flow intensity, $|\Psi|_{max}$. The nearly linear variation of $|\Psi|_{max}$ with $|\Theta_2|$, which is illustrated in figure 28, suggests that circulation is dictated primarily by the magnitude of Θ_2, and is relatively independent of $\Theta_1{}^h$. This is reasonable from a physical standpoint because the influence of $\Theta_1{}^h$, which accelerates fluid in the x_1-direction, is canceled out by the $\Theta_1{}^h$ contribution from the other side of the cavity. This phenomena is manifested by the impingement of fluid in the center which restricts motion in the x_1-direction. It does not apply to flow in the x_2-direction which, apart from the stress conditions along the sidewall and center-line, is less constrained in terms of momentum. The relationship between circulation and Θ_2 also explains why the flow intensity is larger for lower contact angles.
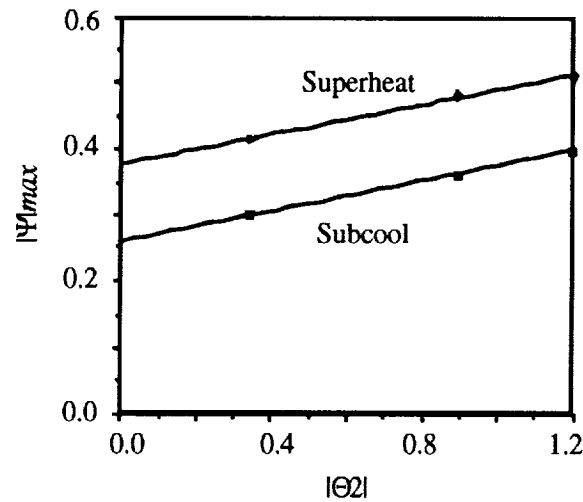
Figure 38. Stress correlation ($Ma = 10^2$ and $Bi = 10$).

As noted earlier, one of the most important consequences of thermocapillary flow with small contact angles is the large dynamic pressure gradients in the contact region. This behavior is illustrated in figure 39 which shows numerically obtained pressure values at the four surface nodes adjacent to the left side wall for the cases depicted in figure 37. It is apparent that dynamic pressure gradient increases dramatically as $\omega \to 0$ and diminishes as $\omega \to 90°$. In fact with a contact angle of 15°, the pressure can be an order of magnitude higher than the value for a flat interface.

Another important aspect is the slightly larger pressure magnitudes for subcooling. This trend appears to contradict the results obtained before with $Ma = 10^3$ in figure 36. At the lower value of Ma ($= 10^2$), the circulation intensity for superheating is only slightly higher than that for subcooling. Hence, the pressure gradient is more sensitive to the local thermocapillary stress exerted at the interline. With subcooling, convection strengthens the temperature gradient, while with superheating, it is weakened. The higher pressure gradient for subcooling is attributed to the larger local stress magnitude in the corners.
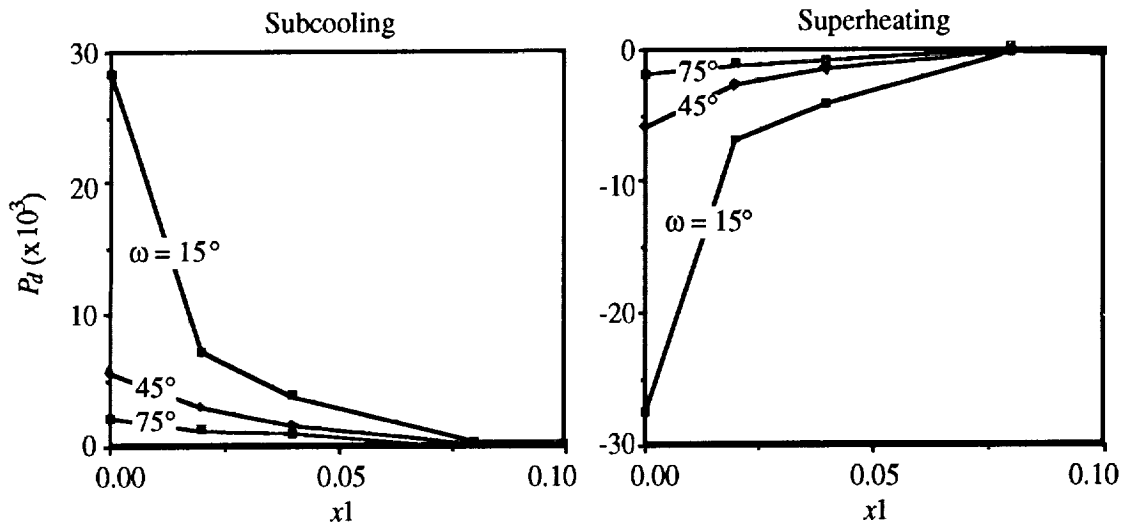
Figure 39. Interfacial pressure near interline ($Ma = 10^2$ and $Bi = 10$).

The effect of Bi and ω on thermal potential and surface stress at the upper limit of Ma ($= 10^3$) is now examined. Figure 40 shows Π/Π_b for subcooling and superheating as a function of ω and Bi. At lower values of ω, the relationship illustrated in figure 34 with $\omega = 15°$ still applies. In fact, the deviation from the basic state increases with larger contact angles. Unlike pure interfacial flow, Π/Π_b for subcooling is greater than unity because thermocapillarity tends to augment meniscus heat transfer by circulating cool liquid from the bottom of the cavity to the middle of the mensicus. However, this effect is significantly less than superheating which yields a much greater increase in Π. With both heating modes, Π/Π_b also increases with Bi due to the stronger circulation intensities and isotherm convection about the surface.

At higher values of ω, however, a drop in Π/Π_b is noted for both heating modes. This is particularly evident for subcooling, but it also applies to superheating at $Bi = 10^2$. With flatter interfaces, the circulation cells in each half-cavity become more constrained by each other. Although the total stress force increases with Bi, the stress concentration shifts to the sidewalls, and the force applied in the center, as a proportion of the total thermocapillary force, decreases. Because of the reduction in the local traction in the center, the flow around the impingement region slows, and

the circulation and temperature deformation become less. This reduction in deformation causes a drop in thermal potential and shifts the maximum deviation in Π to a lower value of Bi. Recall that $\Pi/\Pi_b \to 1$ as $Bi \to \infty$ for all contact angles.
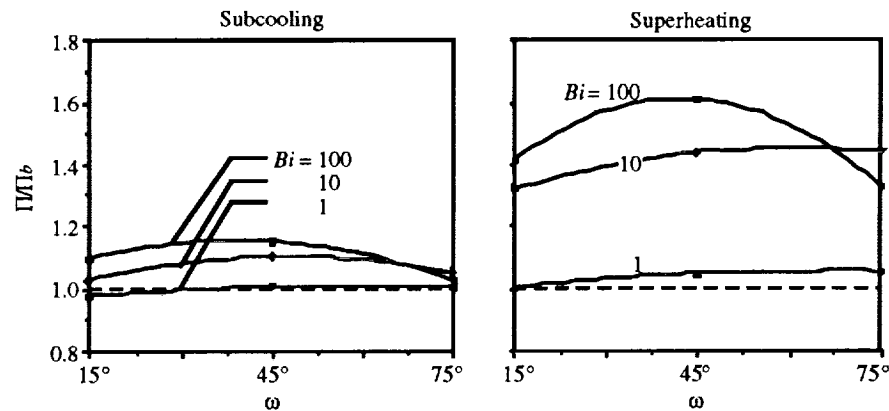
Figure 40. Π/Π_b versus ω and Bi (thermocapillary flow).

The sensitivity of Θ_2/Θ_{2b} to ω and Bi is shown in figure 41. Although the ratio is relatively independent of ω, the variation in Θ_2/Θ_{2b} with Bi is more pronounced for superheating because the thermocapillary convection associated with this regime tends to reduce the temperature gradient and yield a lower modified stress than the basic state. Consistent with the trend observed with pure interfacial flow, $\Theta_2/\Theta_{2b} \to 1$ as $Bi \to \infty$. This is because the deformation becomes less due to the flat temperature distribution in the center of the cavity.

The complex relationship for Θ_2/Θ_{2b} with subcooling is due to competition between (1) the increase and shifting of the temperature gradient towards the sidewalls, and (2) the upwelling of cool fluid from the bottom of the cavity which causes an opposing gradient around the center-line. With the former effect (1), one would expect Θ_2/Θ_{2b} to be consistently greater than 1, which would mirror the trends for superheating. This effect predominates for low Bi and ω, where the influence of upwelling is small. With increased Bi, however, upwelling lowers the average gradient to a value less than the basic state. With increased ω, the gradient shift is almost totally offset by upwelling.

Figure 41. Θ_2/Θ_{2b} versus ω and Bi (thermocapillary flow).

74
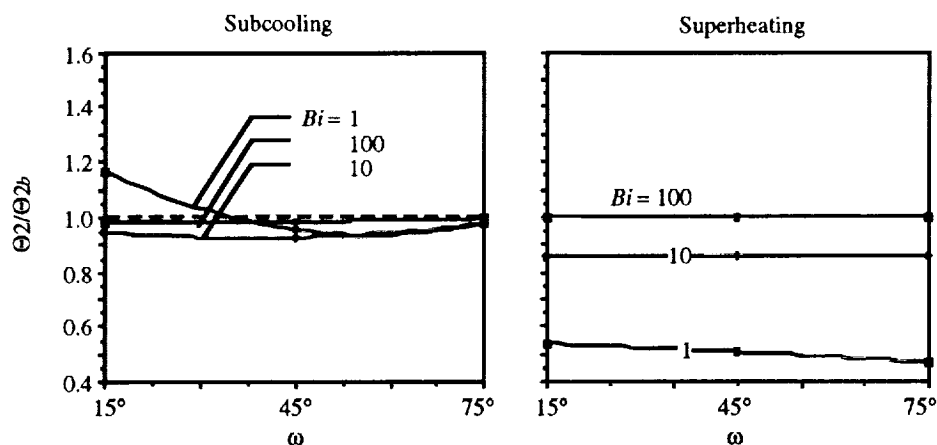
E. Combined Flow Regimes

Since interfacial flow and thermocapillarity yield opposing circulations when subjected to the same heating conditions, the combined regime reflects a competition between two convection modes. The relative influence of each is indicated by the half-cavity circulation C which changes sign according to the predominance of either regime. With combined flow, the circulation, velocity field, and surface temperature distribution are dictated by the relationship between Ma, Rs, and Bi. At low Bi, the surface temperature gradients are small. Consequently, the flow is more sensitive to variations in Rs than Ma, and it manifests many of the features of pure evaporation or condensation. For high Bi, the interfacial temperature gradients are larger, and the influence of interfacial flux is restricted to the interline. The flow in this case becomes less sensitive to Rs, and tends to model pure thermocapillary behavior.

Figure 42 shows examples of combined mode condensation and evaporation for $Ma = 10^3$, $Rs = 10^{-1}$, and $Bi = 10$. With subcooling, the flowfield assumes a circulation pattern similar to its pure thermocapillary counterpart in figure 29, but it also contains negative-value streamlines representing the flow of condensing liquid from the interline region to the bottom of the cavity. This flow tends to shift the vortices on either side of the cavity to the center-line, and causes a distention of surface isotherms towards the bottom, similar to its pure condensation counterpart in figure 18.

From the standpoint of circulation, $C = 46.45$ for the combined regime, which is less than the value of its thermocapillary counterpart ($C = 56.24$). Although the combined regime is thermocapillary-dominant, its $|\Pi|$ value of 0.55 is less than either the pure thermocapillary or condensation cases (which are 0.62 and 0.56, respectively). This is illustrated in figure 42 by the difference between sidewall and center-line stream functions at $x_2 = 0$, i.e., $|\Delta\Psi|$, which reflects mass transfer through the half-cavity. For the combined regime, $|\Delta\Psi| = 2.73$, which is only slightly less than the value for pure condensation, $|\Delta\Psi| = 2.78$.
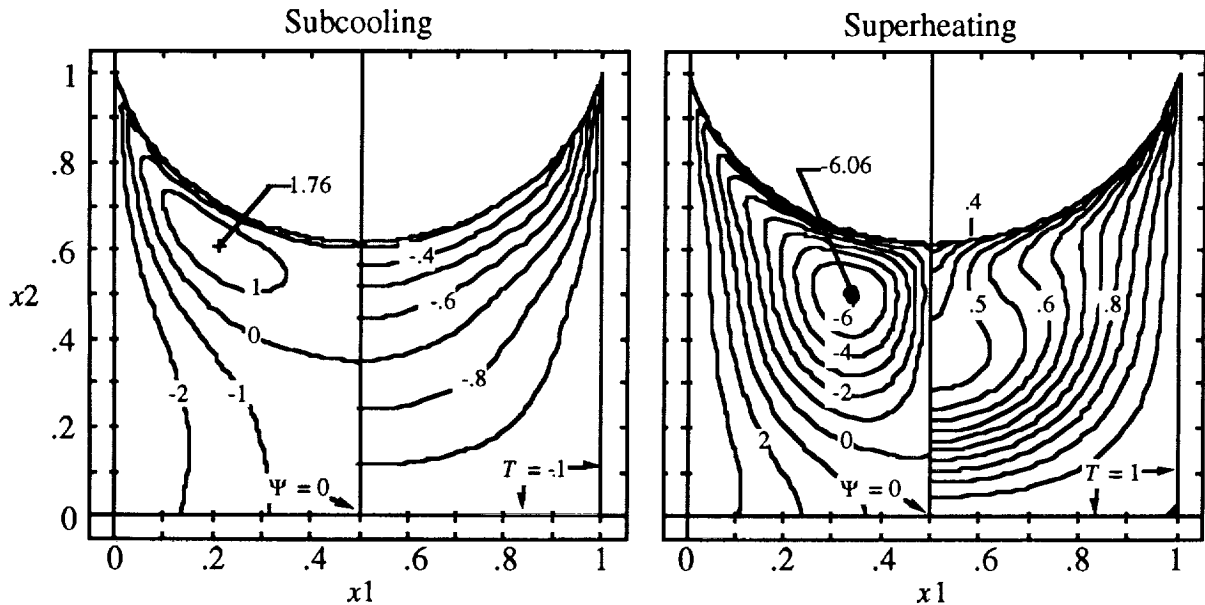
Figure 42. Combined convection ($Ma = 10^3$, $Rs = 10^{-1}$, and $Bi = 10$).

It is clear that the heat transfer for combined convection does not represent a simple average or superposition of thermocapillary and condensation effects. Rather, thermocapillarity modifies the surface temperature in a manner that augments the tendency of condensation to accumulate warm fluid on the surface. This behavior is tantamount to removing the |Π|-increasing influence of upwelling in subcooled thermocapillary flow. In combined mode condensation, the cooler center region that results from upwelling is eliminated by warm condensate in the middle of the cavity. Without this offsetting influence, the sole action of thermocapillarity is to extend this warm region toward the corners, thereby lowering the thermal potential relative to pure condensation.

With superheating, the evaporation flux, which is represented by positive-value streamlines, convects the vortices and liquid toward the corners. The circulation magnitude for this case, $C = -102.31$, is lower than its pure thermocapillary counterpart, $C = -137.85$, but the flow clearly remains thermocapillary-dominant. The thermal potential also remains the same at $|\Pi| = 0.797$. Although the flowfield is thermocapillary-dominant in terms of circulation and thermal potential, the total mass flowrate across the meniscus $|\Delta\Psi| = 3.77$ is greater than its pure evaporation counterpart ($|\Delta\Psi| = 3.49$) in figure 18, and the surface temperature profile promotes evaporation.

The effect of reducing Bi to 1 is illustrated in figure 43. The lower value of Bi reduces the surface temperature gradient and thermocapillary stress. Interfacial flow is also higher because of the increased magnitude of Π. With subcooling, the circulation value of 5.07 indicates a transitional flow structure quite different from its pure thermocapillary counterpart in figure 27, where $C = 16.05$. This difference applies to thermal characteristics, as well. The thermal potential of the combined regime ($|\Pi| = 1.09$) is less than either pure thermocapillary flow ($|\Pi| = 1.18$) or pure condensation ($|\Pi| = 1.13$). In addition, the mass throughput of $|\Delta\Psi| = 5.42$ represents a slight decrease relative to the pure condensation case of 5.59. As in the case of $Bi = 10$, thermocapillarity contributes to the suppression of thermal potential caused by condensation.
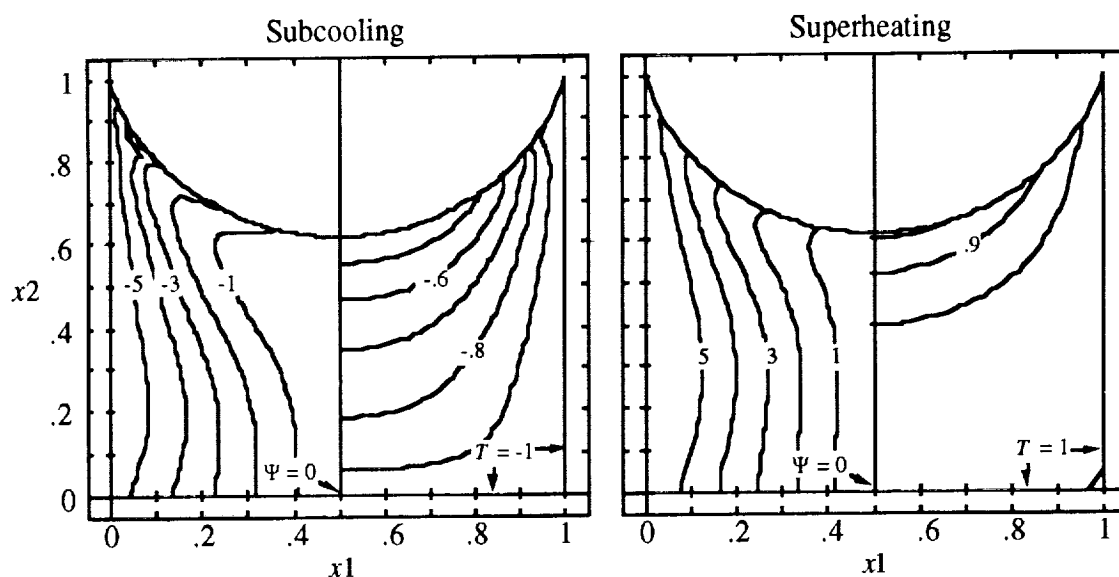
Figure 43. Combined convection ($Ma = 10^3$, $Rs = 10^{-1}$, and $Bi = 1$).

With superheating, the lower value of Bi also yields a circulation ($C = 2.70$) that reflects a transition between pure thermocapillarity and evaporation (where $C = -28.36$ and 19.49, respectively). Although the positive value of C implies a more interfacially dominant regime than

76

combined mode condensation, the thermal potential is basically the same as its pure thermocapillary counterpart, where $|\Pi| = 1.21$. Unlike the case of $Bi = 10$, this value of 1.21 is less than the thermal potential for the pure evaporation case, where $|\Pi| = 1.25$, and is only slightly greater than the basic state value of 1.20. Consequently, the mass throughput for the combined regime $|\Delta\Psi| = 5.84$ is less than the pure evaporation $|\Delta\Psi|$ value of 6.11. The opposing action of thermocapillary circulation restricts interfacial convection and isotherm compression at the surface, thus reducing evaporation.

The influence of reducing Ma is illustrated by comparing figure 44 with figure 43. With subcooling, the circulation change from 46.45 to -4.66 when Ma is decreased from 10^3 to 10^2, respectively, represents transition to a slightly condensation-dominant flowfield. Although the low magnitude of C at $Ma = 10^2$ implies that neither regime is completely dominant, the vortex driven by interfacial stress disappears and is replaced by a vertical flow structure very similar to figure 18. In fact, $|\Pi|$ and $|\Delta\Psi|$ increase to 0.56 and 2.77, respectively, which are the same values as the combined regime's pure condensation counterpart. Reduction of Ma clearly decreases the suppression of thermocapillarity on $|\Pi|$, and the flowfield becomes condensation-dominant in terms of heat and mass transfer.
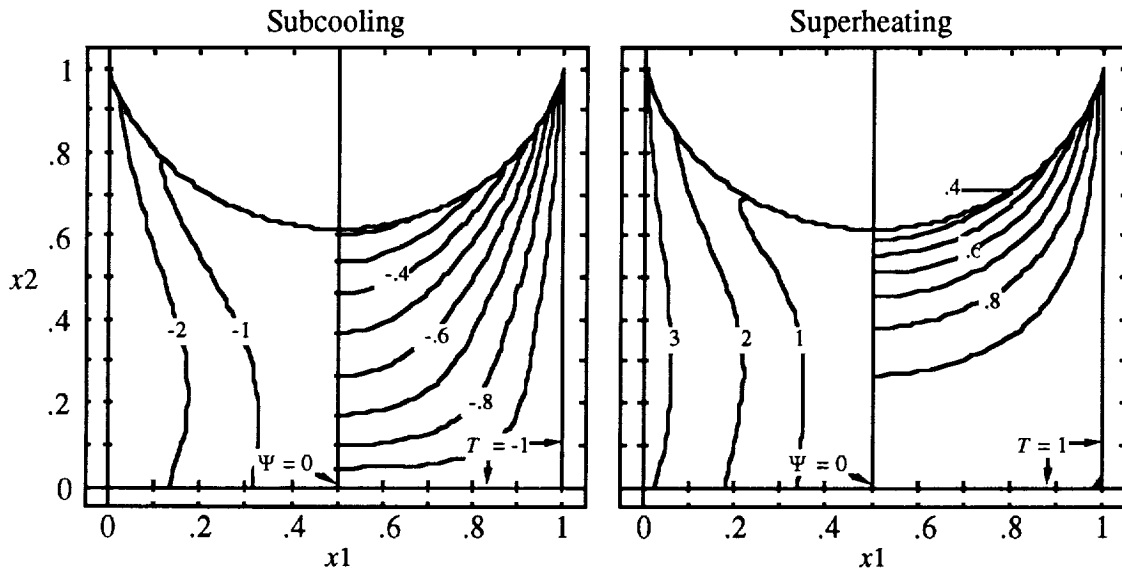
Figure 44. Combined convection ($Ma = 10^2$, $Rs = 10^{-1}$, and $Bi = 10$).

With superheating, the circulation change from -102.31 to 4.00, when Ma is decreased from 10^3 to 10^2, represents a transition to evaporation-dominant flow. Recall that at $Ma = 10^3$, the combined regime enhanced evaporation by raising the thermal potential with respect to pure evaporation (i.e., $|\Pi| = 0.80$ for combined mode, while $|\Pi| = 0.72$ for pure evaporation). Upon lowering Ma, the circulation patterns of both modes offset each other, thereby reducing convective effects near the surface. This is manifested by a drop in $|\Pi|$ to 0.67, which is slightly higher than the basic state value of 0.61 but less than the pure evaporation value of 0.72. It is also manifested by the flowrate $|\Delta\Psi| = 3.13$, which is less than either case in figures 41 and 18. It appears that the convection that drives isotherm deformation relative to the basic state is neutralized as $C \rightarrow 0$.

The effect of raising Rs is shown by comparing figure 45 with figure 44. With both condensation and evaporation, increasing Rs from 10^{-1} to 1 produces a thermocapillary-induced vortex and an approximately order-of-magnitude reduction in interfacial flow. The flow transitions

from a structure dominated by interfacial flow (i.e., C = -4.66 for subcooling and C = 4.00 for superheating) to a thermocapillary-dominant structure (i.e., C = 5.72 for subcooling and C = -7.45 for superheating).
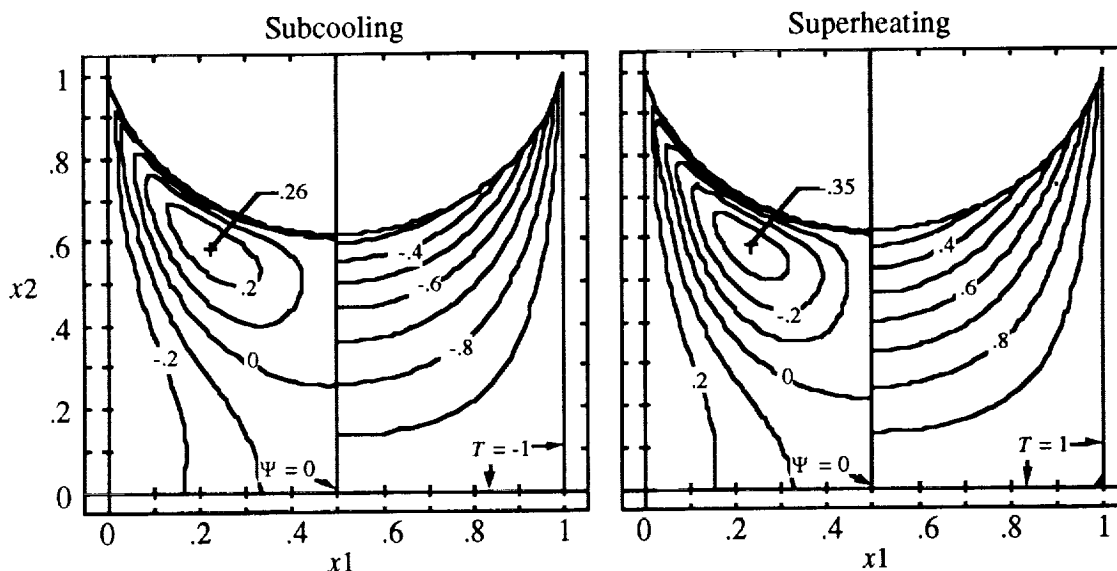
Figure 45. Combined convection (Ma = 10^2, Rs = 1, and Bi = 10).

The thermal potential for subcooling increases from 0.56 to 0.59 due to the diminished accumulation of warm condensate on the meniscus, but it is still less than the $|\Pi|$ associated with the basic state. With superheating, however, $|\Pi|$ drops to a value close to the basic state. Since thermocapillarity and interfacial flux both increase thermal potential, an increase in nonequilibrium (or decrease in $1/Rs$) should reduce the nonlinear influence of Bi on temperature. Note that increasing interfacial resistance from 10^{-1} to 1 drops the half-cavity throughput $|\Delta\Psi|$ from 3.13 to 0.35 for superheating and from 2.77 to 0.29 for subcooling.

The influence of Ma, Rs and Bi is summarized in figure 46 which shows circulation versus Rs for Ma = 10 and 10^2, and Bi = 1, 10 ,and 10^2.

To facilitate comparisons, a corrected circulation C^* is used, where C^* = C for superheating/evaporation and C^* = $-C$ for subcooling/condensation. With these definitions, thermocapillary-dominance is indicated when C^* < 0, while interfacial-dominance occurs when C^* > 0. The transition region is defined to include not only the crossover point at C^* = 0, but also the bounds where C^* changes from being relatively independent of Rs (i.e., Rs < 1) to being strongly dependent on Rs and Bi (i.e., Rs > 3×10^{-2} to 10^{-1}).

At the lower limit of Bi (= 1), thermocapillary flow is relatively weak compared to either condensation or evaporation. The low interfacial temperature gradient yields a very low circulation intensity in the thermocapillary-dominant region. The influence of Ma in this case is vanishingly small as illustrated by the negligible difference in C^* between Ma = 10 and 10^2. For this range of Ma, the transition regime occurs between 0.5 < Rs < 10. At lower values of Rs (i.e., Rs < 0.1) interfacial flow clearly dictates circulation intensity.

78
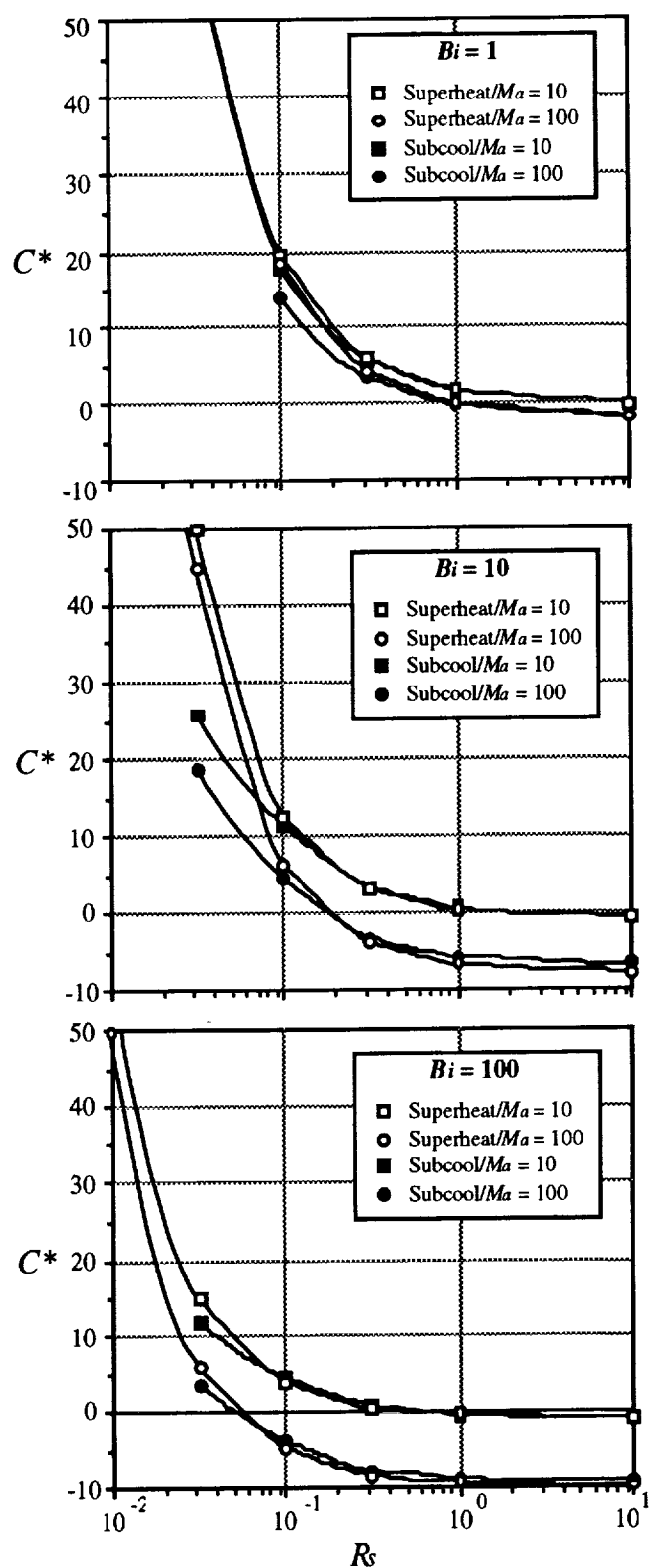
Figure 46. Half-cavity circulation versus Ma, Rs, and Bi.

With superheating, the influence of Ma on C^* essentially vanishes for $Rs < 10^{-1}$, and the flow becomes solely a function of Bi and Rs. The increasing accumulation of warm liquid as $Rs \to 0$ reduces the interfacial temperature gradient and diminishes the influence of Ma. This behavior, however, does not apply to subcooling, which assumes a lower C^* and distinct Ma dependency for $Rs < 10^{-1}$. As mentioned before, the lower C^* is due to the suppressing influence of condensation on $|\Pi|$. This reduction in $|\Pi|$ is complemented by an associated increase in $Bi\Theta_2$ and thermocapillary stress. Thus, the differences in Ma persist at low Rs, and tend to augment the suppression of $|\Pi|$ at larger condensation rates.

At higher Bi (= 10), the increased temperature gradient shifts the transition to interfacial dominance to lower values of Rs. In addition, the values of Rs representing the transition region are influenced much more by Ma. In general, the point at which C^* becomes independent of Ma occurs at a lower value of Rs. At the lower limit of Ma (= 10), the transition region shifts slightly to the left due to the very small circulation intensity. At higher Ma (= 10^2), the shift to a range of $0.03 < Rs < 0.3$ is more noticeable due to the increase in thermocapillary circulation strength. Although it cannot be discerned from the plot, one expects C^* for $Ma = 10$ and 10^2 to converge as $Rs \to 0$.

Another aspect of figure 46 is the different slopes exhibited by the subcooling and superheating curves in the transition region. As discussed previously, this behavior is due to the opposite influence of evaporation and condensation on interfacial temperature. With superheating and evaporation, interfacial temperature and $|\Pi|$ increase steadily with lower Rs due to isotherm compression. With subcooling, however, $|\Pi|$ is reduced due to distension of isotherms towards the bottom of the cavity. The Rs value at which this behavior becomes noticeable tends to decrease at higher Bi. At $Ma = 10^2$, the crossover point occurs near the transition $C^* = 0$. This means that over the entire range of Rs and Bi, C^* for superheating is always greater than subcooling. This is not the case at the lower limit of Ma. Here, it appears that there are values of Rs at which C^* is greater for condensation.

A further increase in Bi to 10^2 yields the same trends as before. Here, however, thermocapillarity is even greater as reflected by the lower transitional ranges. The transition from thermocapillarity to interfacial flow at $Ma = 10$ occurs at $Rs \approx 0.5$, while at $Ma = 10^2$ it occurs at $Rs \approx 0.05$. The temperature suppression of condensation is also more evident and creates a distinct disparity in the C^* values for subcooling and superheating in the interfacially dominated region. With this combination of parameters it appears that the influence of thermocapillarity persists for $Rs \leq 10^{-2}$ since the curves do not seemingly converge. However, it is expected that the convergence noted before will occur as $Rs \to 0$.

The influence of Rs, Bi, and heating mode on thermal potential at $Ma = 10^2$ is shown in figure 47. With superheating, both thermocapillarity and evaporation augment one another in terms of their influence on $|\Pi|$. Since the Marangoni number examined here is too small to effect appreciable convection-induced deformation of the temperature field, the value of $|\Pi|$ for $Rs \geq 1$ (thermocapillary-dominant regime) is essentially constant and equivalent to the basic state value. For $Rs < 1$, however, $|\Pi|$ increases relative to the basic state due to the compression of isotherms towards the surface. One also sees that the increase in $|\Pi|$ is greatest at a Bi close to 10. This is consistent with trends for pure evaporation in figure 23, which indicated a maximum increase between $1 < Bi < 10$. With condensation, the surface potential in the thermocapillary-dominant region still closely approximates the basic state because of the small isotherm deformation. With reduced Rs, however, $|\Pi|$ decreases due to the previously mentioned increase in surface temperature.
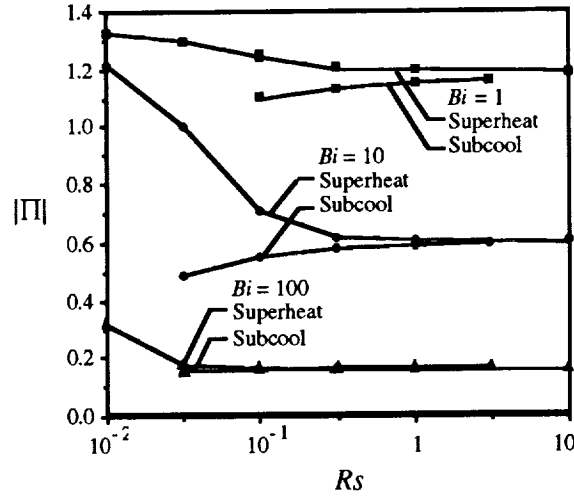
Figure 47. |Π| versus Rs and Bi (combined convection).

In section VI.D, it was found that circulation and interfacial pressure gradient were very sensitive to contact angle. A similar analysis of combined mode condensation and evaporation is performed by examining contact angles of 15°, 45°, and 75°, while holding $Ma = 10^2$, $Rs = 10^{1/2} = 0.31623$ and $Bi = 10$. The corresponding stream function and isotherm plots for these cases are illustrated in figure 48, and a summary of important flow parameters is shown in table 6. A comparison between tables 6 and 5 indicates that the influence of ω on pure thermocapillary flow and combined convection is generally the same. For instance, |Π|, |C| and $|P_d|_{max}$ are greater at small contact angles, and |C| for all values of ω is consistently higher for superheating. With combined-mode subcooling, the suppression of upwelling by condensation results in a |Π| lower than either the pure thermocapillary or condensation cases. Alternatively, with superheating the mutually augmenting effects of evaporation and thermocapillary stress yield higher thermal potentials than with pure thermocapillary convection.

The most notable difference between the convection modes in tables 5 and 6 is the substantially higher pressure gradient associated with combined flow. Numerically obtained values at the four nodes adjacent to the sidewall are shown in figure 49. Although the circulation magnitudes in table 6 are approximately 40 to 50 percent less than their counterparts in table 5, the values of $|P_d|_{max}$ are between 1.5 to 2.5 times higher than pure thermocapillary flow. The difference is greatest for subcooling with $\omega = 15°$ in which the presence of condensation raises $|P_d|_{max}$ by a factor of 2.47. This contrasts with the smallest increase which occurs with superheating at $\omega = 15°$. Here, $|P_d|_{max}$ increases by a factor of 1.43.

Regardless of the contact angle or heating mode, combined convection yields a higher interfacial pressure gradient than either its pure thermocapillary or interfacial counterparts. This is because both modes yield similar pressure distributions near the interline, and the combination of the two represents a superposition of their distributions. For instance, the vertical flow structure of pure condensation yields a positive pressure gradient in the x_2-direction. Because the fluid is wetting, the interline pressure must be higher to compensate for its increased distance from the base of the cavity. With pure subcooled thermocapillary flow, the interline pressure is also higher to accommodate the high momentum flux through this region. In both cases, the interline pressure is greater at lower contact angles.

81
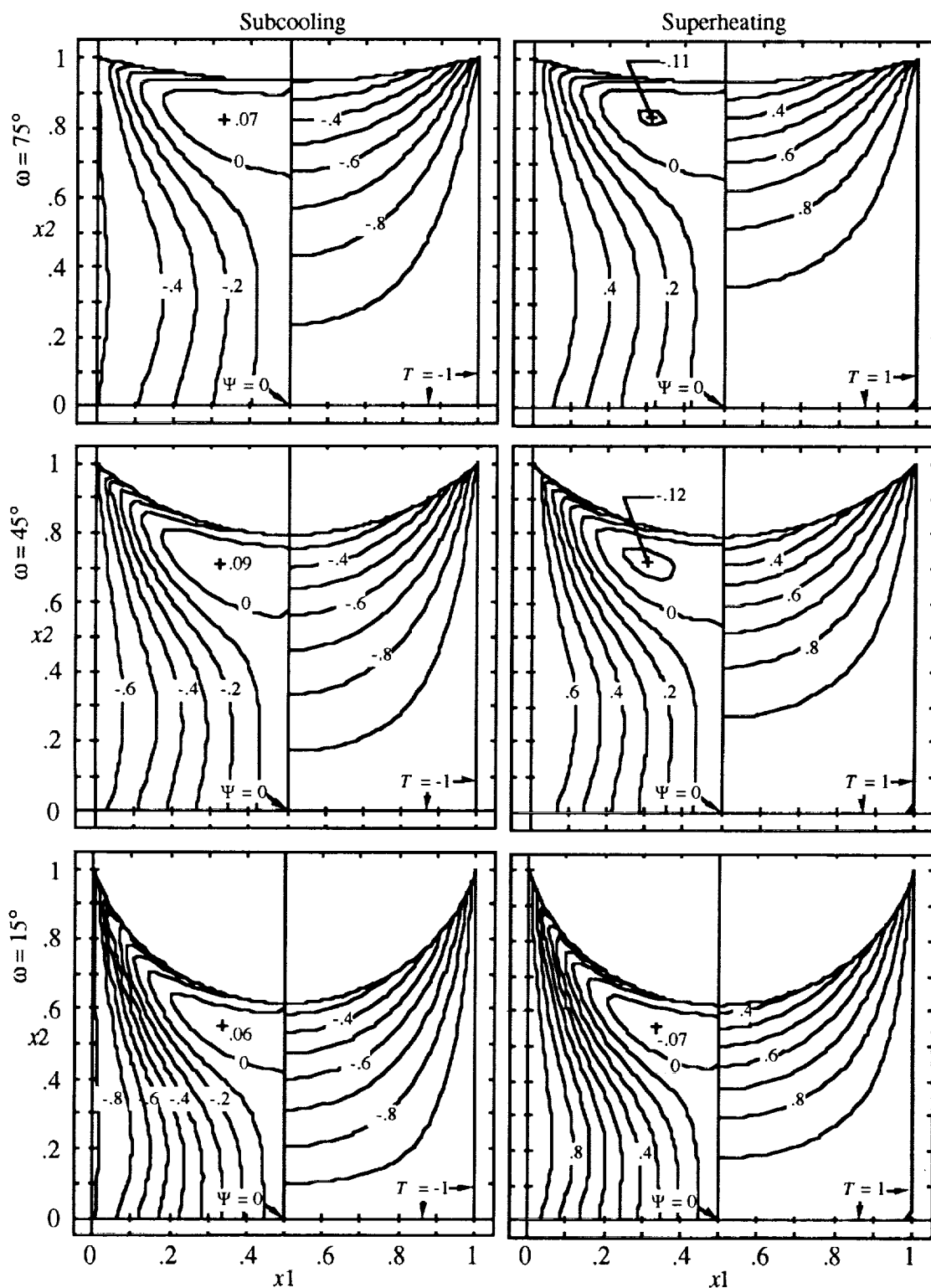
Figure 48. Contact angle sensitivity ($Ma = 10^2$, $Rs = 10^{-1/2}$, and $Bi = 10$).

C-2

Table 6. Pore Characteristics ($Ma = 10^2$, $Rs = 10^{-1/2}$, and $Bi = 10$).

Mode	ω	$|\Pi|$	$|Bi\Theta_1{}^h|$	$|Bi\Theta_2|$	$|\Psi|_{max}$	C	$|P_d|_{max}$
Subcool	15°	0.576	0.419	1.233	0.06	3.30	69,930
	45°	0.403	0.621	0.925	0.09	2.81	11,600
	75°	0.321	0.784	0.354	0.07	1.73	4,247
Superheat	15°	0.620	0.419	1.166	0.07	−3.20	39,490
	45°	0.445	0.621	0.872	0.12	−3.23	9,447
	75°	0.359	0.773	0.334	0.11	−2.02	3,162
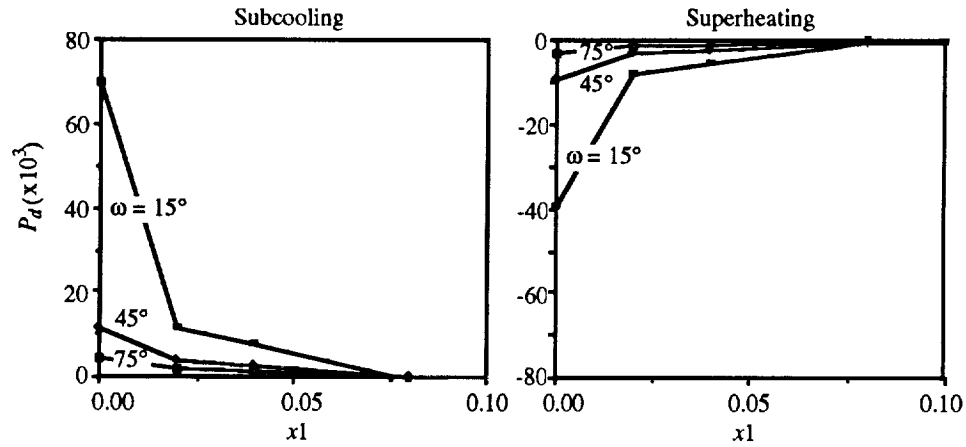
Figure 49. Interfacial pressure near interline ($Ma = 10^2$, $Rs = 10^{-1/2}$, and $Bi = 10$).

F. Summary

In this section, the steady-state thermophysical behavior of a two-dimensional pore subjected to superheated and subcooled boundaries was examined. The free surface was assumed to be circular and wetting, and was treated solely as a function of contact angle ω. For analysis of the basic state, several parameters were derived which characterize the nonlinear influence of Bi and ω on interfacial transport and thermocapillary stress, namely thermal potential Π and modified stress θ_i. It was shown that integrating θ_i along the surface yields a net Re-normalized traction, $Bi\Theta_i$, which acts in either the $+x_2$ (subcooling) or $-x_2$-direction (superheating). Due to the relative orientation between the surface and internal temperature gradient, this force increases as $\omega \to 0$ or $Bi \to \infty$, but vanishes as $\omega \to 90°$ or $Bi \to 0$. It was also found that $|\Pi|$ is larger for smaller contact angles, but decreases as $Bi \to \infty$.

Pure evaporation and condensation establish vertically oriented flowfields in the cavity. The flowfield strength is characterized by the mass throughput and circulation in each half-cavity, and, for a given ω, Bi and Rs, is always greater for evaporation. With evaporation, the positive deviation in Π from the basic state reflects the convection of heated liquid towards the surface, which strengthens

and reinforces interfacial transport. With condensation, the deviation is negative due to the accumulation of warm condensate which suppresses mass flow into the cavity. The interfacial temperature increase associated with evaporation flattens the temperature profile and reduces the stress force relative to the basic state. The temperature rise associated with condensation, however, causes a peaking about the center-line which increases stress.

Pure thermocapillary flow is characterized by a pair of counter-rotating cells about the center-line. The strength of each cell is characterized by the half-cavity circulation which tends to increase in magnitude with larger Ma and Bi. The circulation arising from thermocapillarity is opposite in sign to that for interfacial flow. In addition, the magnitude is always less for subcooling due to this regime's higher viscous losses along the side walls. It was found that superheating increases interfacial temperature and Π relative to the basic state by convecting heated liquid from the sidewalls towards the center-line. With subcooling, a more complex behavior arises from the competition between surface convection and upwelling. At low circulation intensities, thermocapillarity extends the heated region along the surface and reduces $|\Pi|$. At higher circulations, however, this effect is increasingly offset and eventually overwhelmed by cool liquid from the bottom of the cavity which causes a net increase in $|\Pi|$. This increase is always less than the positive deviation associated with superheating.

Another important result was the increase in circulation intensity with lower contact angles and proportionality of cell strength to the x_2-component of the thermocapillary stress force. It was also found that smaller contact angles yield significantly larger dynamic pressure gradients along the surface and magnitudes at the interline. With subcooling, pressure increases towards the sidewall, and assumes a positive value with respect to the base of the cavity. With superheating, the gradient is negative, and yields a large negative value at the contact point.

The flowfield for combined-mode convection reflects a competition between thermocapillary and interfacial flow. With superheating, the circulations for evaporation and thermocapillarity are opposite, but both modes promote thermal potential and suppress the surface stress force. Consequently, the Π values in the thermocapillary-dominant regime yield higher evaporation rates than pure interfacial flow. In the transitional region, however, the opposing circulations for thermocapillarity and evaporation suppress convective effects, and yield surface temperatures approximately equivalent to the basic state. In the interfacially dominant regime, the flow and Π behave the same as in pure evaporation. The influence of thermocapillarity and Ma vanishes with decreasing Rs because of isotherm compression and reduction of the surface temperature gradient.

The behavior of subcooling is complicated by the tendency of condensation to promote surface stress. In the thermocapillary-dominant region, the flow is very similar to superheating, but the positive deviation in Π is much less. In the transitional regime, two important effects occur that cause a departure in behavior. One is the reduction in thermal potential and restriction on condensation rate caused by accumulation of hot liquid around the meniscus center-line. The other is the reinforcement of thermocapillary flow due to the increase in temperature gradient. Because of these effects, the transition to an interfacially dominant flow structure occurs at a lower value of Rs than superheating. In addition, the sensitivity of circulation to Ma persists in the interfacial-dominant regime because of condensation's reinforcing effect on thermocapillary convection.

It was also found that the contact angle can influence the transition between thermocapillary and interfacially dominated flow and that the flowfields at lower ω tend to be more thermocapillary-dominant. Another important observation was the dramatic increase in interfacial pressure gradient

84

at low ω. This occurs because the combined-mode pressure distribution represents a superposition of the pressure fields associated with each mode, which share the same general distribution.

VII. SURFACE DEFORMATION

A. Introduction

In sections III and V, the causes of surface deformation were discussed, and in section IV, a method for solving meniscus geometry was developed. In the steady flow analysis of section VI, however, this report followed the approach of other researchers and neglected deformation of the free surface by assuming vanishingly small Capillary and Crispation numbers. This assumption would be valid except for several important differences between the problem addressed in this study and those performed before.

First, the length-scales considered here are extremely small (i.e., 1 μm $< D < 10^2$ μm), and the scaling analysis has shown that Ca, which varies inversely with dimension, can be significant for $D < 10^3$ μm. Secondly, comparisons of the interfacial pressure distributions in section VI indicated a dramatic increase in the magnitude of P_d near the interline at low contact angles. Most prior studies of thermocapillary flow with deforming interfaces considered liquid with high contact angles ($\omega \sim 90°$) or pinned end locations. The surface pressure gradients for these cases, as verified in section VI, were probably small. With low contact angles, however, the pressure gradients may be substantial enough to cause significant deformation, even at moderate to low values of Ca. A third major difference is that previous work in thermocapillary flow concentrated on either combined thermocapillary/buoyancy-driven convection or thermocapillarity alone. Hence, the added influence of vapor recoil on float-zone type surfaces has not been adequately studied.

In the interest of limiting computational time, it is desirable to neglect surface deformation, since it adds another iteration to the steady-state solution. This is especially true if it has only a second-order effect. However, if it occurs to even a moderate extent, then it should be considered for several reasons. First, in a system whose operation depends on sustaining a pressure difference between a vapor and liquid, such as a screened LAD, extreme meniscus deformation could result in retention loss. The normal stress balance in equation (45) includes several terms that compete in dictating interface curvature. With certain parameter values, a combined flowfield/surface solution may diverge in the iteration process, thereby indicating a numerical or physical instability.

In addition to altering the exposed surface area for heat transfer, a deforming interface may also yield temperature distributions substantially different than the static nondeforming state. Depending on the magnitude of the variation, this may cause a large deviation in heat transfer, which would have ramifications for a system involving phase change across a porous interface.

Before attempting simultaneous solution of the flowfield and surface, it is desirable to estimate the approximate deformation associated with each stress term in equation (45) and the flow regimes covered in section VI. This points to the conditions leading to mechanical non-equilibrium of the surface, and enables a more definitive comparison of terms in the normal stress equation. The approach involves using the interfacial velocities, pressures and temperatures

obtained for the nondeforming cases in section VI to calculate new surfaces based on the numerical integration procedure described in section IV.C.

This approach is equivalent to assuming that the flowfield and temperature are independent of surface geometry. Although it is probably adequate for small deformations (i.e., $Ca \approx 0$ and $Cr \approx 0$), such an assumption would lead to erroneous results if the deformation was high. If the first-order evaluation indicates significant deformation at values of Ca, Cr, and Vr within the ranges identified in the section V, then the coupled solution must be considered. In addition, extreme deformation in the form of surface inflection points could represent sources of mechanical nonequilibrium.

B. Surface Tension Dependence

Surface tension variation along a nonisothermal meniscus will contribute to deformation not only through the convection arising from thermocapillary stress but also the adjustment of capillary pressure to changes in temperature. The latter effect is manifested by the $(1-Cr\,\overline{T})^{-1}$ term in equation (45), where Cr represents the sensitivity of surface tension to temperature. Warmer regions of the surface, $T > T_l$, will assume larger curvatures $\kappa\ (= \partial\alpha/\partial s)$ to offset the reduction in surface tension. The conditions imposed by Cr are evinced by expressing γ in terms of Cr and γ_l. Rearranging and substituting equation (43) into equation (26) yields:

$$\gamma = -\frac{Cr\gamma_l}{|\Delta T|}(T - T_1) + \gamma_l \ . \tag{124}$$

At the maximum temperature, $T = T_h = |\Delta T| + T_l$, which when substituted into equation (124) yields:

$$\gamma = \gamma_l(1 - Cr) \ . \tag{125}$$

For a normal fluid in which $\partial\gamma/\partial T < 0$, Cr must be less than unity to ensure that $\gamma < \gamma_l$. Having $Cr > 1$ represents a situation where $\partial\gamma/\partial T > 0$ and $\gamma > \gamma_l$. A limitation of the linear equation of state for γ, equation (26), and definition of Cr becomes obvious when $Cr = 1$. In this case, the maximum temperature limit T_h corresponds to the critical temperature, and with superheating or very high Bi, γ vanishes and $\kappa \to \infty$ at some point on the surface. Thus, the condition of $Cr \approx 1$ indicates a need for either a higher-order approximation for γ or a critical state model of the surface. The parameter bounds from section V, however, indicate that the value of Cr is low enough to justify use of a linear approximation. In addition, it will be shown that the influence of Cr only becomes apparent for $Cr > 0.9$, which is well above the limits expected for a normal fluid.

The first-order influence of Cr is examined by applying a simplified version of equation (110) in which all terms, besides $(1-Cr\,\overline{T})^{-1}$, are neglected. Since the only factor causing deviation from a circular profile is $\overline{T} = \overline{T}(x_1)$, the relationship is:

$$\kappa = \frac{\partial\alpha}{\partial s} = \frac{\Sigma}{1 - Cr\overline{T}} \ . \tag{126}$$

Employing the approach outlined in section IV.C, one calculates contact angle estimates by integrating along the surface and varying Σ until the contact angle condition at $|x'| = 0.5$ is satisfied. Surface tension-induced deformation is first examined using the basic state interfacial temperature distributions ($Ma = 0$ and $Rs = \infty$). Half-cavity meniscus contours for $Bi = 1$, 10 and 10^2, and $Cr = 0.5$, 0.9, 0.99 and 0.999 are shown in figure 50. The responses to the two heating modes are opposite. With subcooling, the temperature at the center-line ($x' = 0$ in meniscus-centered coordinates) is a maximum and decreases to the minimum T_l at the side walls ($|x'| = 0.5$), that is $\partial T/\partial x' < 0$. Because of the lower surface tension in the middle of the cavity, the equilibrium surface assumes a higher curvature relative to the side wall region. At low Bi, the effect is small, even at high Cr, because temperature remains close to the minimum limit along the entire surface. At high Bi, the temperature in the center approaches the maximum $\overline{T} \to 1$ and the surface becomes more sensitive to Cr. Although the deformation appears to be quite large at high Bi, it is still effectively negligible for $Cr < 0.5$.

With superheated boundaries, $\partial T/\partial x' > 0$, and temperature decreases towards the center-line. Unlike subcooling, the condition of maximum curvature occurs at the side wall since γ is a minimum in this region. Additionally, the higher surface tension in the center requires a lower curvature to maintain stress equilibrium. The net result is a flattening and raising of the meniscus with respect to the contact line at $|x'| = 0.5$. At low Bi, the temperature remains close to the side wall value along the entire surface and is more sensitive to variations in Cr. At higher Bi, however, the lower average temperature yields a surface that is less sensitive to Cr.

The influence of convection due to thermocapillary stress, evaporation, and condensation on γ-induced deformation reflects the departure of interfacial temperature from its basic state distribution. As shown in figures 21 and 22, evaporation and condensation both increase the average temperature of the interface. This tends to reduce the average temperature gradient for evaporation but increases it for condensation.

The effect of pure interfacial flow on surface geometry is shown in figure 51 for $Bi = 1$ and $Rs = 10^{-1}$. Although the differences are quite small, condensation appears to increase deformation relative to the basic state, while evaporation decreases it. The surface is also more sensitive to Cr for both heating modes because of the increase in average temperature. Although it is not shown, the deformation at high Bi ($= 10^2$) is the same as the basic state, because the temperatures for all three regimes are essentially equivalent. Note that the influence of Cr is negligible for all cases in which $Cr < 0.9$.

The sensitivity of surface morphology to pure thermocapillary flow with $Ma = 10^3$ and $Bi = 10$ is shown in figure 52. Unlike the previous cases, the thermocapillary convection associated with subcooling tends to lower interfacial temperature relative to the basic state, while superheating raises it. Consequently, the subcooling regime exhibits less sensitivity to Cr than its basic state counterpart in figure 50. The deformation associated with superheating, however, behaves very similarly to the basic state at $Bi = 10$.
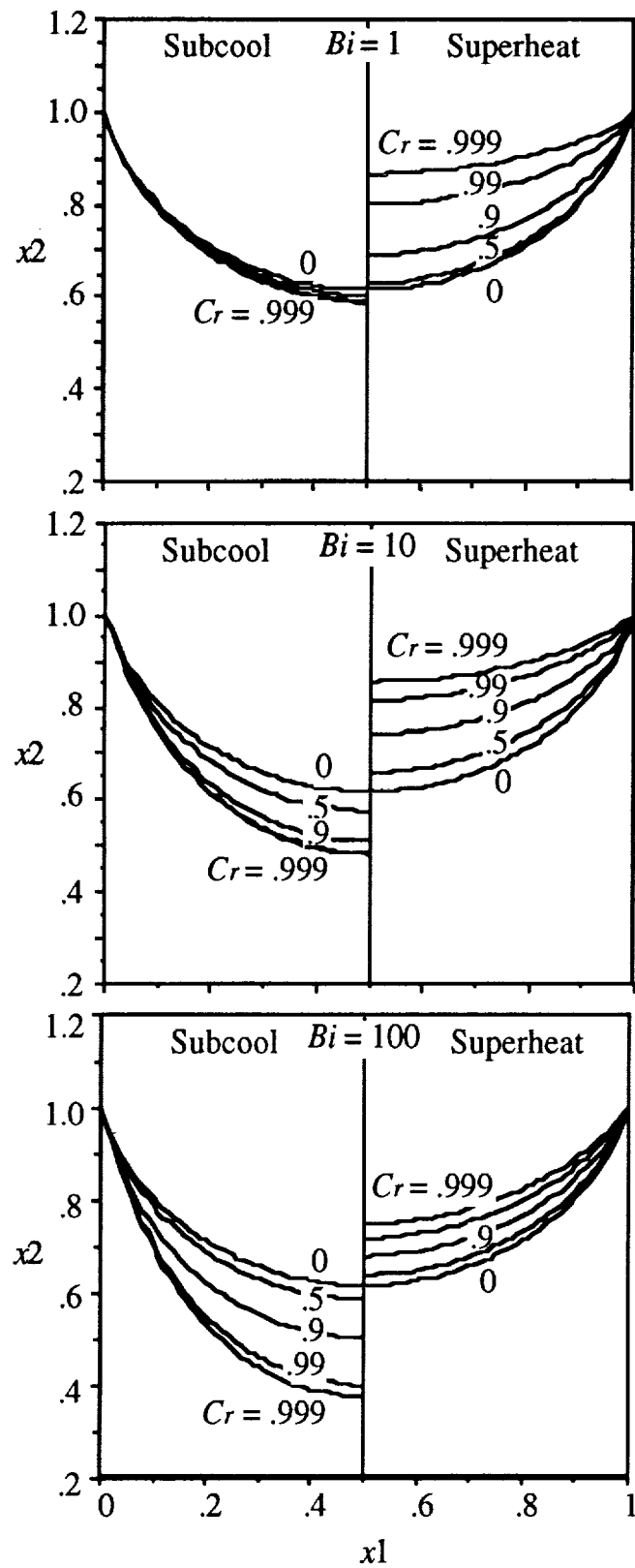
Figure 50. Surface tension sensitivity (basic state temperature).
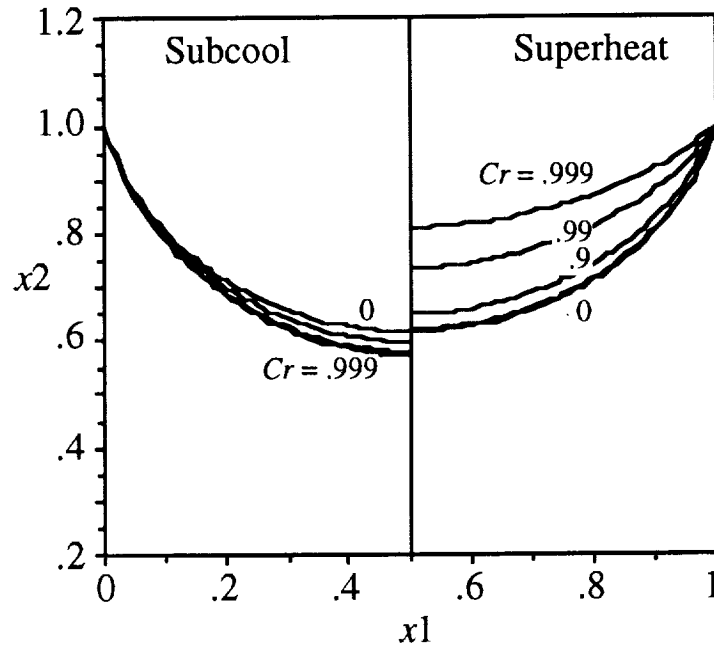
Figure 51. Surface tension sensitivity ($Ma = 0$, $Rs = 10^{-1}$, and $Bi = 1$).

C. Vapor Recoil Dependence

The term "vapor recoil" refers to the momentum change exerted on the meniscus due to liquid/vapor phase change. The recoil force is represented by the VrT^2 and $Vr\Delta(T^2)$ terms in equations (45) and (110), respectively.
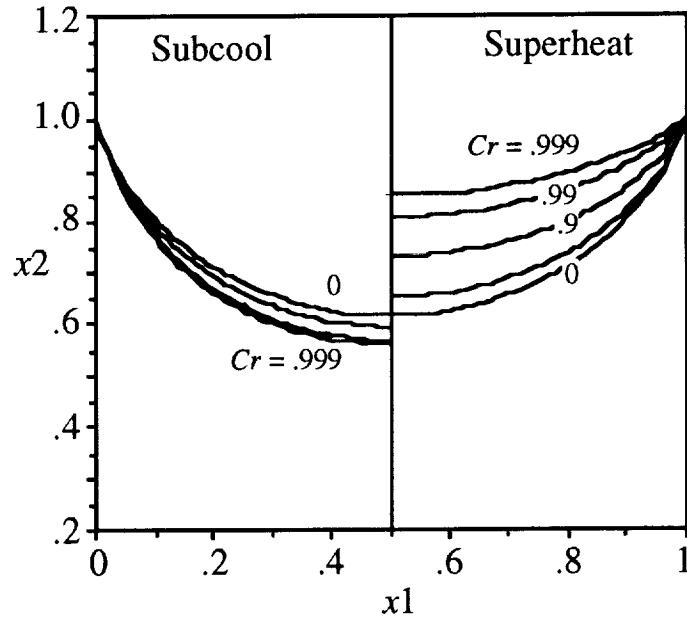
Figure 52. Surface tension sensitivity ($Ma = 10^3$, $Rs = 10^3$, and $Bi = 10$).

For both heating modes, recoil varies with the square of the temperature difference, and is always positive. Regardless of the direction of mass transfer, it tends to raise surface curvature towards the side walls and yields a force pointing into the liquid. From the jump mass balance, we know that $\rho_v(V_i n_i)_v = \rho_l(V_i n_i)_l$. Because $f_\rho = \rho_l/\rho_v \gg 1$, the magnitude of the vapor momentum flux, $P_v = j(V_i n_i)_v$, is always greater than that of the liquid, $P_v > P_l$, regardless of their relative directions. With evaporation, the difference between momentum entering and exiting the surface is negative $P_l-P_v < 0$, and yields a net force opposing the direction of flow. With condensation, the net momentum entering the surface is P_v-P_l. Since each of these terms is negative, $P_v-P_l < 0$, and the net force again points into the liquid.

The effect of recoil is examined using another simplified version of equation (112), namely:

$$\kappa = \frac{\partial \alpha}{\partial s} = \Sigma + Vr\Delta(T^2) .$$
(127)

As before, the basic state temperature distribution is employed as an initial reference. Although this case lacks the influence of internal convection, it does enable a consistent comparison with section VII.B. Figure 53 shows equilibrium surface geometries for $Bi = 1$, 10 and 10^2. The temperature magnitudes for superheating and subcooling are equivalent and should yield the same recoil force distribution. The disparity between the subcooling and superheating curves in figure 53 are attributed to slight differences in the steady-state numerical convergence.

Ignoring these discrepancies, one sees that over the entire range of Bi, raising Vr tends to reduce curvature in the middle of the cavity and flatten the surface relative to the contact line. Since $\partial(\Delta(T^2))/\partial x' > 0$, the recoil contribution to $\partial\alpha/\partial s$ increases toward the side wall. That is, the curvature near the side walls must be higher to compensate for the increased normal force directed into the liquid. At high Vr, the center-line curvature can become negative and form an inflection along the meniscus.

It is evident from figure 53 that the sensitivity to Bi is maximized between 1 and 10^2. This is due to the manner in which $\Delta(T^2) = T^2-T_0^2$ varies along the surface. At the lower limit of $Bi (= 1)$, interfacial temperature, mass flux and $\Delta(T^2)$ follow a moderate variation and increase towards the side wall. The surface assumes a negative curvature in the center to accommodate the 15° contact angle constraint and the increasingly positive contribution of $\Delta(T^2)$. At higher $Bi (= 10)$, this variation is more pronounced, thus causing a larger change in curvature. Above a certain value of Bi, the interfacial temperature assumes a flat distribution in the center followed by a sharp increase or decrease at the interline. This is reflected in the plot for $Bi = 10^2$. Near the center-line, the temperature is low and fairly constant, and the contribution to curvature is negligible $\Delta(T^2) \approx 0$. Although $\Delta(T^2)$ rises sharply in the side wall region, its influence on overall curvature is slight. Hence, the deformation diminishes compared to the lower values of Bi.

The influence of thermocapillary convection is illustrated in figure 54. For these plots, the surface temperatures corresponding to $Ma = 10^3$ and $Rs = 10^3$ is used. Although interfacial flow was ignored in these steady-state solutions, the temperatures still reflect the general influence of thermocapillary convection.
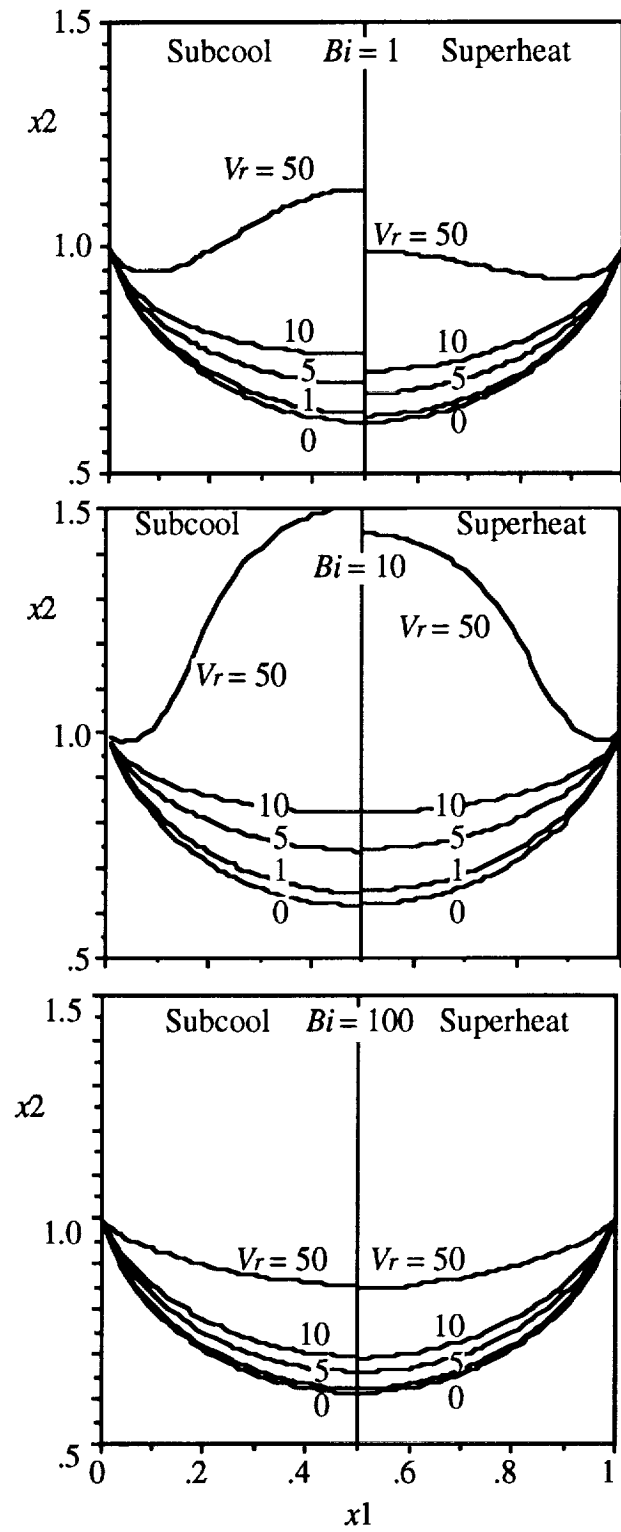
Figure 53. Recoil sensitivity (basic state temperature).
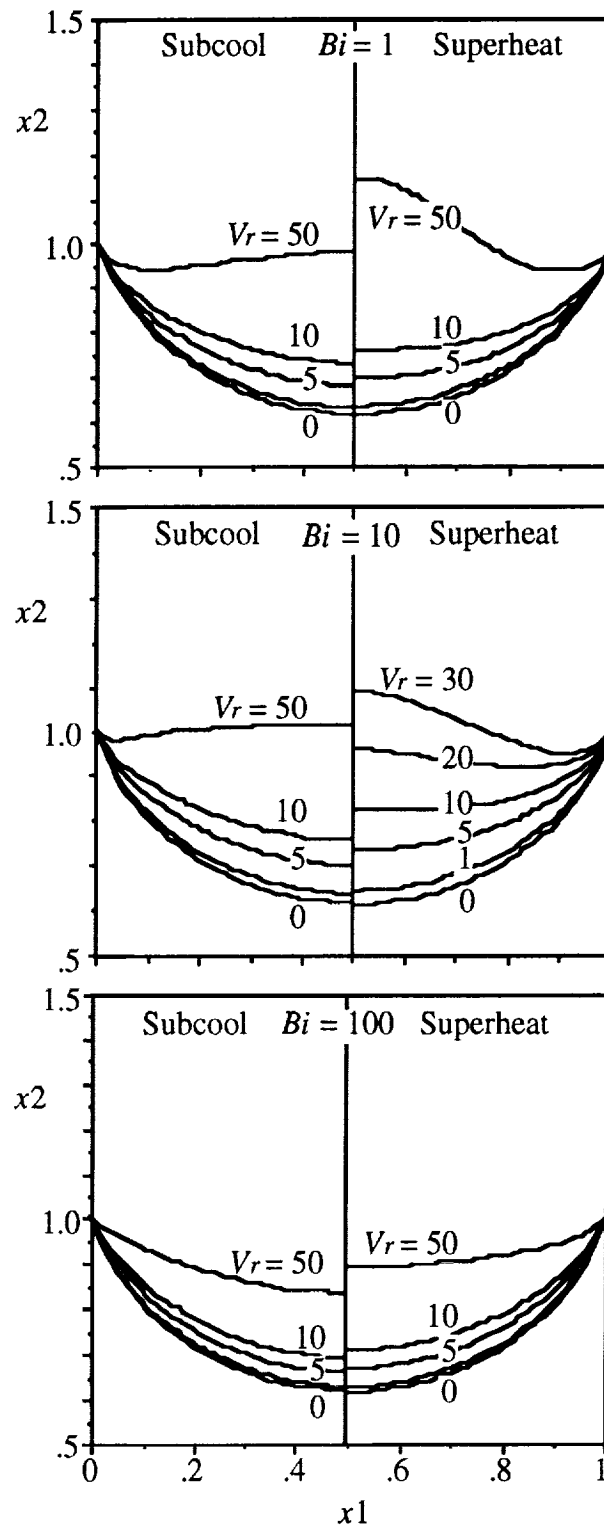
Figure 54. Recoil sensitivity ($Ma = 10^3$ and $Rs = 10^3$).

With subcooling, thermocapillary stress shifts the temperature gradient toward the side wall and flattens the temperature distribution in the center of the cavity. This is similar to increasing Bi in that the meniscus becomes less sensitive to recoil. One also sees that the deformation at $Bi = 10$ is much less than the basic state, because of the temperature depression associated with upwelling. This phenomenon yields a situation where the recoil contribution decreases from the center-line to the point of maximum surface temperature. This trend reverses beyond the temperature maximum and causes a sharp upturn in the surface near the side wall. At the upper limit of $Bi (= 10^2)$, deformation is similar to the response based on the basic state in figure 53.

Figure 54 shows an opposite trend with superheating. The difference is due to the temperature trough in the middle of the cavity that arises from convection along the surface. The value of $\Delta(T^2)$ at the center-line is consequently higher and causes a large change in curvature in the middle of the pore. At low $Bi (= 1)$, the deformation is much greater than the basic state. At higher $Bi (= 10)$, the deformation is large enough that a single-valued solution at $Vr = 50$ does not exist. The inability to account for a multivalued surface is due to the limitation of the numerical procedure, in which T is expressed as a function of x_1 rather than s. In cases where the surface loops over itself between $0 < |x'| < 0.5$ to satisfy the acute contact angle constraint at $|x'| = 0.5$, the approach of expressing $T(x_1)$, $P_d(x_1)$, etc. is completely inaccurate. However, it is likely that such configurations are inherently unstable and susceptible to pinching off near the points where $dy^{(s)}/dx' \rightarrow \infty$ and $d^2y^{(s)}/dx'^2 > 0$. At the upper limit of $Bi (= 10^2)$, the influence of the temperature trough diminishes and the deformation closely approximates behavior of the basic state-based response.

The sensitivity of deformation to pure condensation and evaporation is shown in figure 55. Unlike the basic state behavior in figure 53, the response to each heating mode is quite different. With condensation, accumulation of warm liquid around the center-line increases the temperature gradient substantially. This, in turn, yields a larger change in ΔT^2 and α from $|x'| = 0$ to $|x'| = 0.5$, and more deformation than the basic state. The change in ΔT^2 and α is related to the modified stress sensitivities shown in figure 24.

With condensation, the increase in Θ_2 relative to the basic state is highest at low Bi and decreases as $Bi \rightarrow \infty$. This trend is reflected in figure 55 by the deformation for $Bi = 1$ being higher than that for $Bi = 10$. Although this appears to contradict the trends in figure 53, the deformation at $Bi = 10$ is still much higher than its basic state counterpart. Consistent with figure 24, one sees that the deformation at $Bi = 10^2$ is essentially the same as the basic state distribution.

With evaporation, the temperature gradient decreases with respect to the basic state. This is illustrated in figure 24, which shows that $\Theta_2/\Theta_{2b} < 0$, and $\Theta_2/\Theta_{2b} \rightarrow 1$ as $Bi \rightarrow \infty$. The consequence of this is that at low $Bi (= 1)$, the deformation is considerably less than its basic state counterpart. With higher $Bi (= 10)$, surface deflection is greater, but is still restricted by the flattened temperature profile. At the upper limit of $Bi (= 10^2)$, the greatly diminished convection yields a surface equivalent to the basic state-based response.
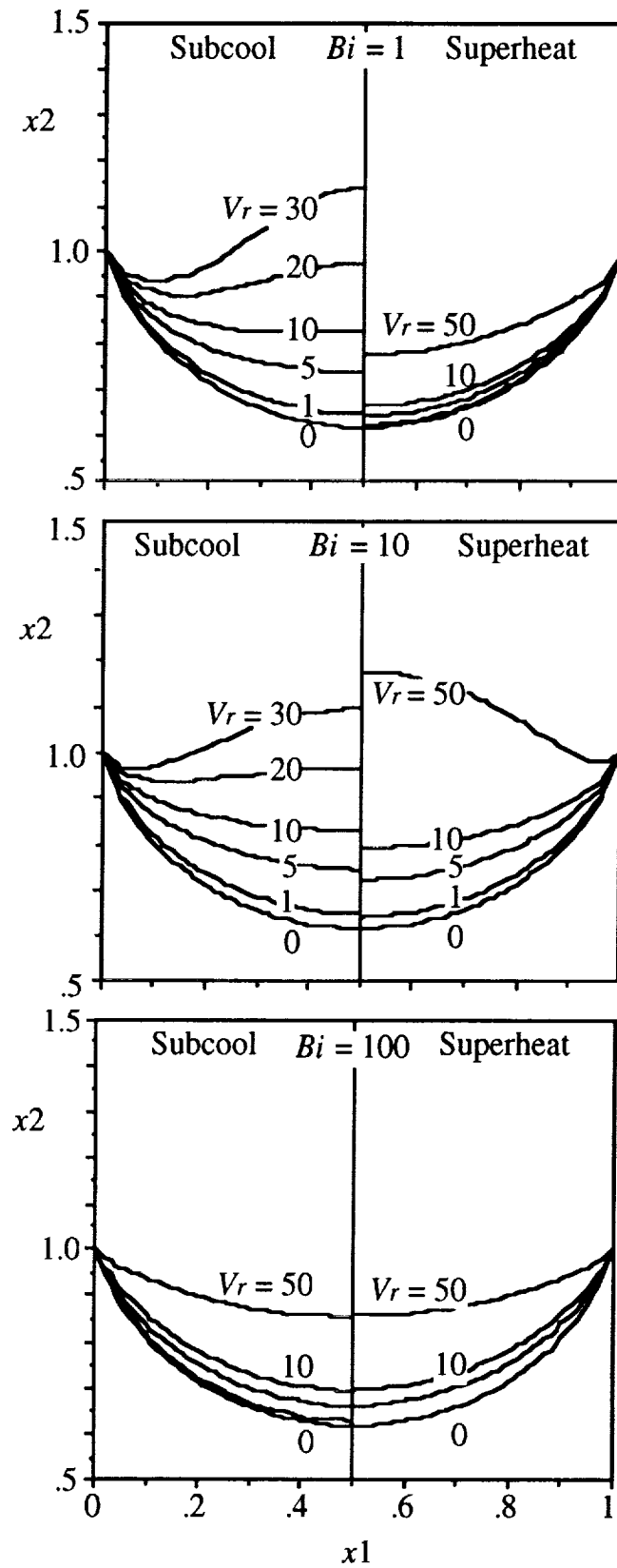
Figure 55. Recoil sensitivity ($Ma = 0$ and $Rs = 10^{-1}$).

D. Dynamic Pressure Dependence

In chapter VI, it was shown that fluid motion can yield substantial pressure variations along the free surface. This pressure P_d falls directly out of the Boussinesq approximation and is best viewed as a deviation from hydrostatic pressure. The influence of P_d on surface morphology is manifested by the $Ca\Delta P_d$ term in equation (110). Ca arises as the dimensionless coefficient for P_d due to our application of viscous scales. This is advantageous because it permits a direct comparison with the effects of vapor recoil and viscous stress. $Ca\Delta P_d$ is expressed as a difference to emphasize that the solution depends on pressure change along the surface and not the absolute value per se. This form underscores the fact that the P_d contribution at the center-line is included in the initial curvature estimate, and that the change in curvature as $|x'| \rightarrow 0.5$ is dictated by the gradient of P_d along the surface.

The results of section VI showed that the pressure variation becomes significant at low contact angles. This was illustrated in figures 36 and 39 which showed that in all cases the pressure magnitude is relatively small in the center of the cavity but increases dramatically at the side walls.

To assess the first-order influence of P_d, one employs the following simplification of equation (112):

$$\kappa = \frac{\partial \alpha}{\partial s} = \Sigma - Ca\Delta P_d \ . \tag{128}$$

The sensitivity to P_d is examined using the pressure distribution for pure thermocapillary flow in which $Ma = 10^3$ and $Rs = 10^3$. The magnitude of the pressure variation, as illustrated in figure 36, increases dramatically at the side walls and becomes more pronounced at high values of Bi. The influence of this change on surface geometry is shown in figure 56, which shows equilibrium surfaces for $Bi = 1, 10$, and 10^2.

With subcooling, the pressure increase towards the side walls is manifested as a suction that pulls the surface down in the middle of the cavity. The magnitude of the deformation becomes larger with increasing Ca, and, like the effect of surface tension in section VII.B, causes a depression in the middle of the meniscus. Equation (128) shows that a positive P_d gradient translates to an increasingly negative contribution to curvature. Since the center-line curvature is positive, this can yield an inflection in the surface at a high enough value of Ca. This inflection is evident near the side wall region of the curve corresponding to $Bi = 1$ and $Ca = 1.5 \times 10^{-4}$. It occurs close to the side wall, since the pressure is comparatively much smaller along the rest of the surface. Another interesting aspect of the inflection is that it causes a negative curvature at the contact line. It was found to be difficult to obtain a solution for Ca greater than this limit. As with recoil, the surface becomes multivalued at a point close to the side wall above a certain value of Ca.

The surface is much more sensitive to Ca at higher Bi because of the larger pressure gradient. This is evidenced by the surfaces in which $Ca = 3.5 \times 10^{-5}$ and 2.0×10^{-5} for $Bi = 10$ and 10^2, respectively. It also appears that the magnitude of the center-line depression depends on the total pressure change from $x' = 0$ to $|x'| = 0.5$, while the location of the inflection leading to a multivalued condition depends on the actual gradient. That is, a more uniform change in pressure will shift the inflection point away from the side wall.
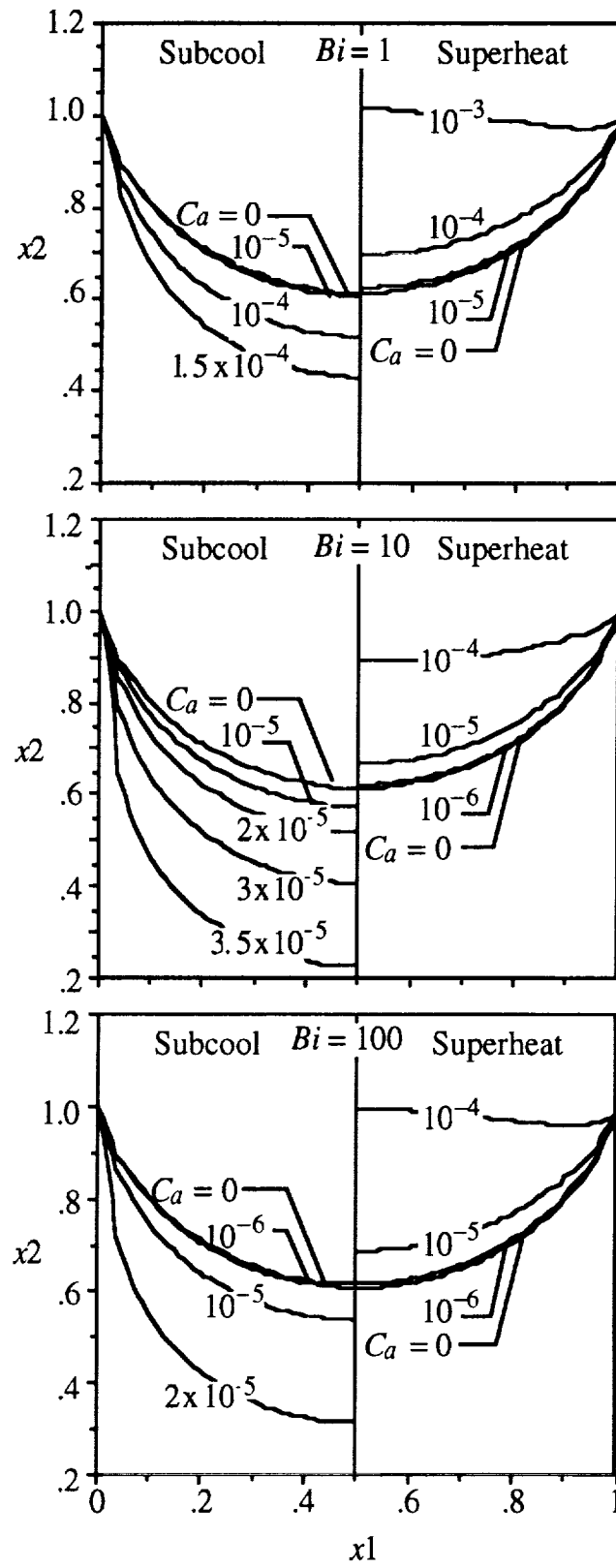
Figure 56. Pressure sensitivity ($Ma = 10^3$ and $Rs = 10^3$).

These trends have important ramifications for the mechanical equilibrium of surfaces with zero contact angles. When $\omega = 0$, the surface is already at a multivalued limit on the side wall. Hence, any stress-related effect that provides a negative contribution to curvature will be impossible to accommodate using the solution procedure in this study.

The deformation in response to superheating reflects a positive contribution to curvature, and behaves similarly to recoil. Here, the negative dynamic pressure gradient causes a suction at the side walls that raises the meniscus center relative to the interline. From a numerical standpoint, the positive contribution to curvature requires a reduced and possibly negative curvature in the center. At low Bi ($= 1$), the meniscus contains an inflection at $Ca = 10^{-3}$, which is nearly an order of magnitude higher than the maximum Ca identified for subcooling. In addition, this value does not appear to be close to the multivalued limit, as in the case of subcooling. Although the overall pressure gradient associated with superheating is greater than subcooling, the surface can support a larger change in contour angle and higher values of Ca before the multivalued limit is reached.

The sensitivity to Ca increases by an order of magnitude from $Bi = 1$ to $Bi = 10$ and 10^2. In all of the superheating cases, most of the curvature change occurs near the side wall, while along the rest of the surface, it remains fairly constant. As a consequence, the inflection point, as with subcooling, lies close to the side wall. We also note that at very low Ca ($<10^{-5}$), the meniscus deformation is rather slight and the magnitude of variation is the same for subcooling or superheating. At higher Ca, the surface for subcooling is much more sensitive to pressure variation.

The first-order surface response based on the steady-state solution for combined thermocapillary/interfacial flow ($Ma = 10^3$, $Rs = 10^{-1}$ and $Bi = 10^2$) is shown in figure 57. As noted before, the surface pressure distribution reflects a superposition of superheated thermocapillary flow and evaporation for the superheated case, and subcooled thermocapillary flow and condensation for the subcooled case. Consequently, the side wall gradients for these regimes are greater and yield an increased sensitivity to Ca.
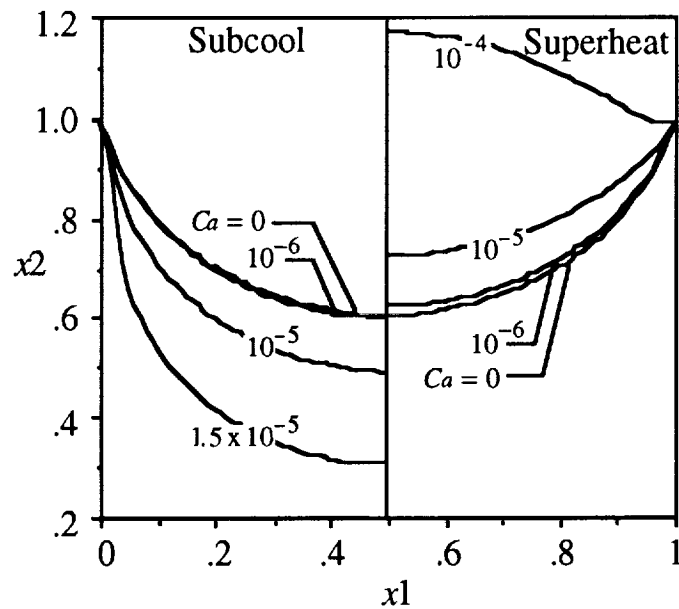
Figure 57. Pressure sensitivity ($Ma = 10^3$, $Rs = 10^{-1}$, and $Bi = 10^2$).

E. Viscous Stress Dependence

The effect of viscous stress on surface morphology is represented by the $2CaV_{i,j}n_jn_i$ term in equation (45). Previous studies have applied it directly in this form, thus requiring calculation of the velocity gradient normal to the surface. With the integral method used here, it is preferable to transform the term into an expression that does not involve the velocity gradient. This improves accuracy and separates the two components of viscous stress which occur only when the surface is curved and volatile. One first employs the chain rule and expands viscous stress into the following form:

$$V_{i,j}n_jn_i = (V_in_i)_{,j}n_j - V_in_jn_{i,j} . \tag{129}$$

$(V_in_i)_{,j}n_j$, is the normal gradient of the mass flux. Note from the equations for j and interfacial equilibrium (equations (7) and (40), respectively), that $V_in_i = T/Rs$. Taking the normal gradient of this relationship and expressing T_jn_j in terms of equation (54) yields the final result which equates the flux gradient to temperature:

$$(V_in_i)_{,j}n_j = -SvT . \tag{130}$$

SvT is termed the "flux component" and represents stress arising from flow normal to the surface. Sv is the so-called viscous stress parameter and is defined as $Sv = Bi/Rs$. The normal vector product of the $V_in_jn_{i,j}$ term in equation (129) can be converted by $n_jn_{i,j} = -s_is_{j,j}$, and substituted with equation (130) into equation (129) to yield:

$$V_{i,j}n_jn_i = -SvT + V_is_is_{j,j} . \tag{131}$$

$V_is_is_{j,j}$ is the so-called "curvature component." With a flat nonvolatile surface, both components vanish and remove the influence of viscous stress. For instance, $1/Rs = 0$ and $Sv = 0$ for a nonvolatile surface. With a flat interface, n_i and s_i are constant, and the condition $s_{j,j} = 0$ must hold. Although the $V_is_is_{j,j}$ term in equation (131) only makes a contribution when the surface is curved, it can also vanish if the tangential velocity is negligibly small, and the flow is normal to the surface.

The flux component is expressed in a form suitable for the numerical procedure described in section IV. However, the unit vectors in the curvature component can be expressed more conveniently by expressing the tangent vector in terms of contour angle. That is, $s_i = \cos \alpha \delta_{i1} + \sin \alpha \delta_{i2}$ and $s_{j,j} = (-\sin \alpha \, \delta_{j1} + \cos \alpha \, \delta_{j2})\partial \alpha/\partial x_j$. Substituting these equations into $V_is_is_{j,j}$ yields:

$$V_is_is_{j,j} = (V_1 \cos \alpha + V_2 \sin \alpha)\left(\frac{\cos^2 \alpha - \sin^2 \alpha}{\cos \alpha \sin \alpha}\right)\frac{\partial \alpha}{\partial s} . \tag{132}$$

One sees that this relation is inappropriate for flat regions in which $\alpha \approx 0$. The value near $x' = 0$ is solved by replacing $\partial \alpha/\partial s$ with $\cos \alpha \partial \alpha/\partial x_1$. Upon expanding the expression and taking the limit as $x_1 \rightarrow 0.5$ (or $x' \rightarrow 0$), one sees that $\cos \alpha \rightarrow 1$ while $\sin \alpha \rightarrow 0$. This leaves an equation that becomes indefinite as $x' \rightarrow 0$, that is:

$$\lim_{x' \rightarrow 0}(V_is_is_{j,j}) = \left(\frac{V_1}{\sin \alpha}\right)\cos^3 \alpha \frac{d\alpha}{dx_1} . \tag{133}$$

Equation (133) is solved by applying L'Hopitals rule and differentiating both numerator and denominator by x_1. After dividing the $\partial \alpha/\partial x_1$ terms out and noting that $\cos \alpha = 1$, one is left with:

$$\lim_{x' \to 0}(V_i S_i s_{j,j}) = V_{1,1} \ .$$
(134)

$V_i s_i s_{j,j}$ becomes important only when the curvature or tangential component of velocity is high. It is the latter condition that distinguishes the viscous stress between volatile and nonvolatile flowfields. Figure 58 compares the surface distributions of $V_i n_j n_{i,j}$ ($= -V_i s_i s_{j,j}$) for pure thermocapillary ($Ma = 10^3$) and pure interfacial flow ($Rs = 10^{-1}$). In the case of thermocapillary flow, the $V_i n_j n_{i,j}$ profiles for superheating and subcooling are opposite in sign because of the different directions of flow. With subcooling, the distribution is flat in the middle of the meniscus, but exhibits a sharp jump and change in sign near the side wall. The magnitude of this jump is more pronounced at higher Bi because of the larger stress gradient and surface velocity. In addition, the convection of the surface temperature gradient and stress toward the side walls causes the magnitude of this peak to approximate the variation in the center. The change in sign near the corners occurs at the point where $\alpha = 45°$ and $s_{j,j} = 0$. For x' above this point, $s_{j,j} < 0$, and for values below it, $s_{j,j} > 0$. The crossover location is the same for all the cases in figure 58, since the surfaces share the same circular geometry.

The variation of $V_i n_j n_{i,j}$ for superheated thermocapillary flow is more pronounced than subcooling. This case also reflects a very different relationship between stress in the middle and corner regions of the surface. The magnitude is higher in the center due to the convection of surface liquid into the interior and concentration of the temperature gradient around the center-line. Consequently, $|V_i n_j n_{i,j}|$ is noticeably larger than the value at the side wall. Although different, the magnitudes in both of these regions are much greater than subcooling because of the higher circulation. With increased Bi, there is an appreciable increase in stress because of the more pronounced temperature gradient around the center-line. At the upper limit of Bi, however, the gradient in the center flattens and the magnitude of $V_i n_j n_{i,j}$ drops off slightly.
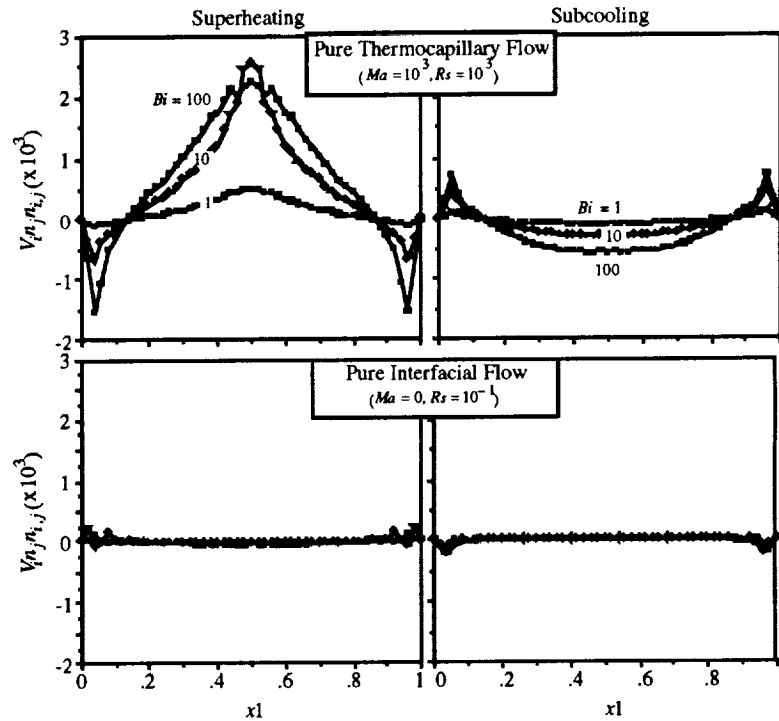
Figure 58. $V_i n_j n_{i,j}$ versus x_1.

As expected, the influence of $V_i n_j n_{i,j}$ for pure interfacial flow is negligible, since the tangential velocity for these cases is extremely small. Except for the small contribution due to the side wall bias of Ψ near the upper corners, the surface velocity is defined primarily by its normal component. Because of the vanishing tangential component, the contribution of curvature to viscous stress can be ignored.

The first-order influence of viscous stress on deformation is analyzed using the same approach from before. The simplified equation for curvature in this case is:

$$\kappa = \Sigma -2Ca(Sv\Delta T - \Delta V_i s_i s_{j,j}) \, , \tag{135}$$

where temperature and velocity are referenced with respect to center-line values according to $\Delta T = T-T_0$ and $\Delta V_i = V_i - V_{0i}$.

The surface response is first examined using the basic state temperature distributions as a reference. Since there is no internal convection associated with this regime, only the influence of SvT can be examined. Figure 59 shows the half-cavity surfaces with subcooled and superheated boundaries for $Ma = 0$, $Rs = \infty$, and $Bi = 1$, 10 and 10^2. The sensitivity is characterized in terms of the product $CaSv$. Unlike recoil, the flux term can assume either a negative or positive deviation from the static isothermal surface, depending on the temperature gradient relative to the center-line.

With subcooling, the higher mass flux near the side walls exerts a force into the liquid that depresses the surface. At low Bi, the deformation is comparatively small since the temperature gradient is low. The surface morphology is most sensitive at $Bi = 10$, since this represents the case of maximum gradient. The deviation drops off at higher Bi because of the shift in temperature gradient towards the side walls. The surface response to the viscous stress flux term is much more sensitive with superheating, as shown by the smaller values of $CaSv$ in figure 59. The difference between subcooling and superheating depends less on convection and more on the numerical robustness associated with terms that add to the curvature in equation (110). With subcooling the flux term is additive, and a large negative curvature at the center-line can be employed to accommodate a large variation in contour angle beyond the inflection point. For instance, with $\omega = 15°$, a total angle change of $165°$ beyond the inflection at the multivalued limit is possible. With superheating, which is subtractive, the angular change is only $15°$.

The flux component for superheating yields a less tractable and more unstable surface than subcooling. On a physical scale, however, the magnitude of this effect is small compared to that of pressure and recoil. For instance at the worst case of $Bi = 10$, an Rs of 10^{-1} and moderate Ca value of 10^{-3} yield $CaSv = 10^{-1}$. The deformation for this case is negligible for both condensation and evaporation.

Figure 60 illustrates the influence of viscous stress using the temperature and velocity distributions for pure interfacial flow, where $Rs = 10^{-1}$. It was noted earlier that because of the negligible tangential velocity, the influence of the curvature component is nonexistent. In addition, the response shown in figure 60 is more relevant than figure 59, from a physical standpoint, since it includes the effects of convection on the temperature field. Here, Sv is fixed at the value appropriate for each case, that is $Sv = Bi/Rs$, and vary Ca. It is evident that the surface is relatively insensitive to Ca, and that the deformation is highest at $Bi = 10^2$. Because of isotherm compression, the gradient with superheating is less thus restricting the influence at $Bi = 10$. It was also noted that the

100

magnitude of deviation is about the same for subcooling and superheating because condensation tends to increase the temperature gradient.
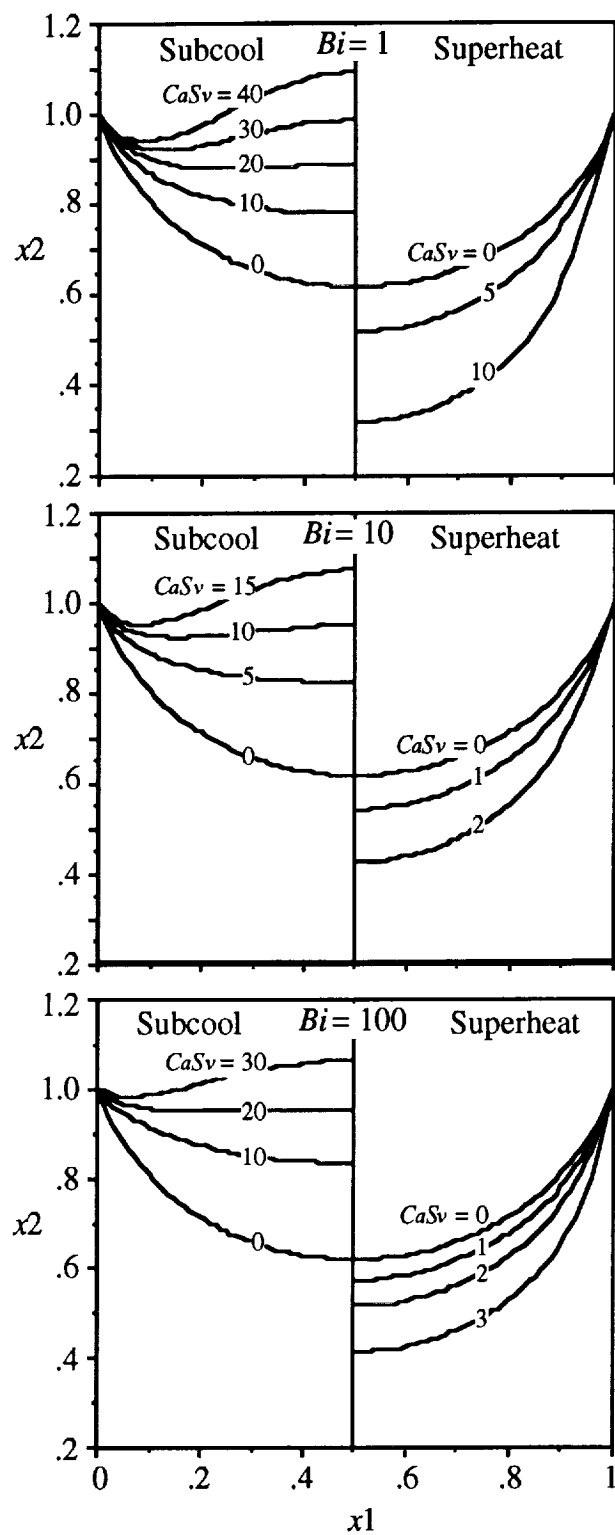
Figure 59. Viscous stress sensitivity (basic state temperature).
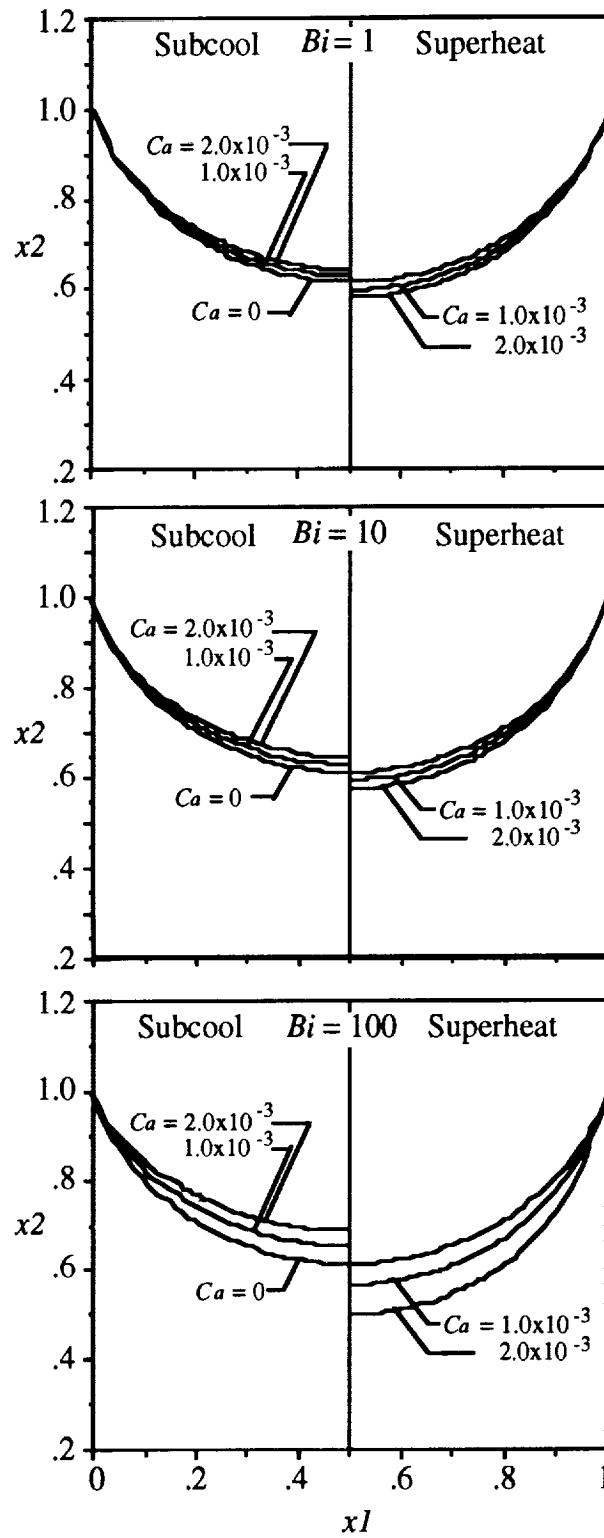
Figure 60. Viscous stress sensitivity ($Ma = 0$ and $Rs = 10^{-1}$).

The response to pure thermocapillary flow is quite different. This case, which is based on the steady-state solution for $Ma = 10^3$, is shown in figure 61. As noted before, the high interfacial resistance effectively removes the flux term from consideration. Thus, the surface is dictated primarily by the change in curvature and $V_i n_j n_{i,j}$. Figure 61 indicates that the deformation response is opposite to that for pure interfacial flow and is more sensitive to changes in Ca. In addition, the deflection increases with higher Bi due to the larger velocities along the surface.
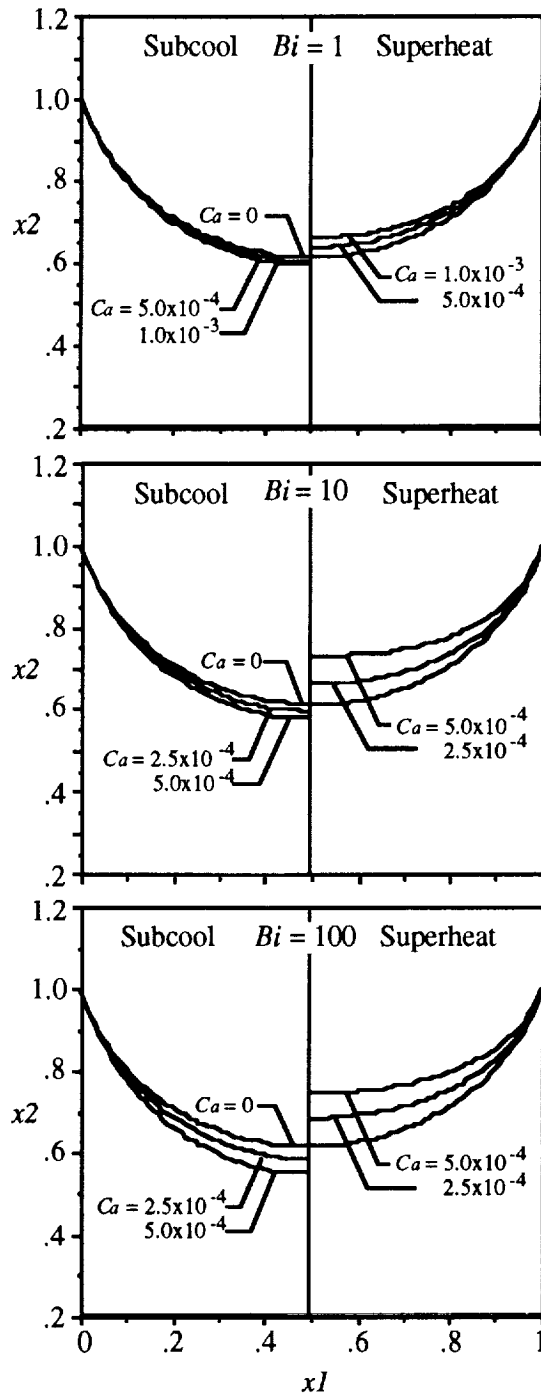
Figure 61. Viscous stress sensitivity ($Ma = 10^3$ and $Rs = 10^3$).

F. Summary

In this section, the different terms in the equation for surface curvature were examined and their first-order effect on surface morphology was evaluated. Results showed that the influence of thermally induced surface tension variation and viscous stress is practically negligible. γ-induced deformation is significant for values of Cr close to unity. This limit, however, is much greater than the bounds identified in section V, and for $Cr < 0.5$, the surface tension term in equation (45) has hardly any influence on surface morphology. The deformation associated with viscous stress is also relatively small. In addition, it appears that the flux and curvature components of this term exhibit opposite behavior and tend to counteract each other in the case of combined convection.

The two most important terms are vapor recoil and dynamic pressure. In general, the influence of recoil is independent of heating mode. It is always manifested in the $-x_2$-direction and increases towards the side walls. This tends to reduce center-line curvature, while raising κ near the contact line. At high Vr, the center-line curvature can become negative, and, if Vr is large enough, lead to a multivalued surface relative to x_1. It was also found that subcooled thermocapillary flow suppresses deformation, while superheating promotes it. These differences, however, vanish as $Bi \rightarrow \infty$. Condensation and evaporation exhibit opposite behavior. Condensation tends to increase deformation by raising the surface temperature gradient, while evaporation suppresses it. Unlike γ-induced deformation, one expects recoil to exert an appreciable influence on surface geometry, since Vr as high as 10 to 10^2 are feasible for Ca ranging from 10^{-4} to 10^{-3}.

The influence of dynamic pressure on surface geometry is quite different for subcooling and superheating. In addition, the large pressure gradients encountered with low contact angles makes the surface extremely sensitive to flow-induced pressure variations and Ca. It is likely that with the pressure distributions obtained in the steady-state analysis, the surface could exhibit multivalued behavior even at low to moderate values of Ca. This is especially true for subcooling which is most sensitive to this effect. Thus, dynamic pressure represents the most likely source of mechanical non-equilibrium.

With superheating, dynamic pressure and vapor recoil act in the same direction. Recoil tends to depress the surface near the side wall, and dynamic pressure does the same by producing a suction in this region relative to the center-line. Both terms make a positive contribution to curvature, and relatively large values of Ca can be accommodated before the surface becomes multivalued. With subcooling, however, the recoil and pressure exhibit opposing effects. Recoil acts in the same direction as before, and depresses the surface near the side walls. Dynamic pressure, however, causes a suction and drawing down of the surface around the center-line. This represents a negative contribution to curvature and greatly reduces the range of Ca that can be accommodated before the multivalued limit is reached.

VIII. PORE FLOWFIELD—DEFORMABLE SURFACE

A. Introduction

The assumption of a frozen or fixed meniscus is oftentimes inappropriate when considering two-phase thermocapillary phenomena in very small pores and cavities. This is especially true for

highly wetting fluids with low contact angles. Results from the first-order analysis in section VII showed that surface morphology is most sensitive to the flow-induced deviation from hydrostatic pressure P_d and the recoil force caused by interfacial mass transfer. It was also found that, within the parameter ranges identified in section V, the surface can acquire an inflection and change in the sign of curvature. Even with a moderate value of Ca ($\approx 10^{-3}$), the inflection can become pronounced enough to form a multivalued surface between the center-line and sidewall.

In this section, the steady-state problem in section VI is reexamined while accounting for a deformable meniscus. The objective in doing this is twofold. From an applications standpoint, the interest is in determining whether the convection and deformation caused by changes in pressurant conditions could lead to retention loss in the pores of liquid acquisition device screens. The flexibility of modeling superheated and subcooled boundaries with different values of interfacial resistance allows us to address the cause of retention loss with heated hydrogen vapor and explain the differences in performance exhibited by helium and hydrogen.

On a general level, the simultaneous influence of dynamic pressure and vapor recoil on surface geometry is investigated, and the extent to which deformation alters the streamfunction, pressure and temperature distributions obtained in section VI is determined. Stability as it is applied in the study of thin films and capillary jets is not addressed. Rather, the surface's ability to maintain mechanical equilibrium is evaluated by determining whether solutions exist that simultaneously satisfy the equations of fluid motion, temperature and surface geometry. Identification of possible metastable states represented by the surface inflections noted in section VII is also of interest. Although these states may be solvable numerically, their liquid geometries are likely prone to slight perturbations in pressure or temperature and indicative of the onset of retention loss.

B. General Effects of Dynamic Pressure and Recoil

Within the context of the one-sided model, the only parameters that directly reflect a change in vapor/liquid equilibrium are Rs and Bi, where $Vr \propto Rs^{-2}$. All others, except contact angle, are fixed at values approximately representative of a 1 μm to 10 μm liquid hydrogen pore subjected to superheat/subcooling levels of 10^{-1} K to 1 K, namely $Ma = 10^2$, $Cr = 10^{-1}$ and $Ca = 10^{-3}$. Instead of considering a zero or very small contact angle, which would properly model hydrogen characteristics, one sets $\omega = 45°$. With subcooling, it was very difficult to obtain solutions at $\omega = 15°$, let alone ~0°, due to the extremely large interline pressure gradients. By employing a higher ω, the pressure gradient is reduced, and the surface is less sensitive to variations in P_d. The advantage of this is that it yields a clearer picture of the deformation associated with dynamic pressure.

First the influence of interfacial equilibrium is examined while holding heat transfer characteristics and all other dimensionless groupings constant. Rs and Vr are varied to model different levels of vapor/liquid equilibrium along the surface, but the sensitivity of latent heat transport to mass transfer is ignored by assuming a constant Bi (= 10). These conditions were examined before in the contact angle investigations of figures 37 and 38. Although the cases illustrated in these figures were solved assuming a constant meniscus, they will nonetheless provide a good reference for examination of deformation. First the focus is restricted to thermocapillary effects by considering the nonvolatile case in which $Rs = 10^3$. After that, progressively lower values of Rs (and higher Vr) are examined to determine how extensively recoil either augments or offsets the influence of dynamic pressure.

For the nonvolatile case, one sees from equation (48) that the extremely low value of recoil parameter (i.e., $Vr \approx 10^{-6}$ with $f_\rho = 10^3$) suggests that the surface is dictated by dynamic pressure and unaffected by recoil. The steady-state solutions for this case are shown in figure 62.
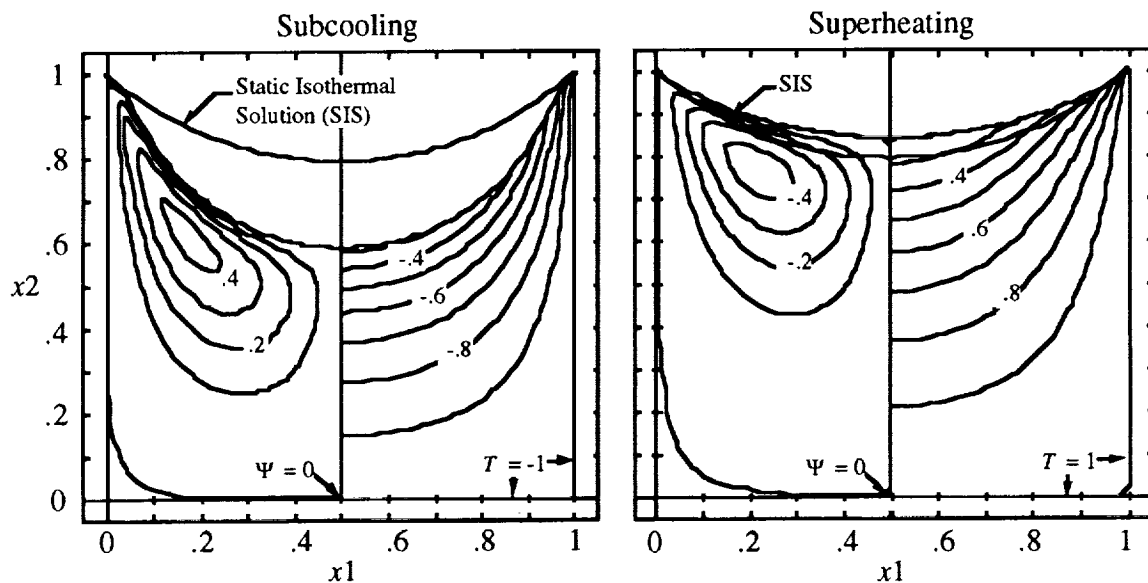
Figure 62. $Ma = 10^2$, $Rs = 10^3$, $Bi = 10$, and $Cr = 10^{-1}$.

To better illustrate deformation, the static isothermal solution (SIS) for the $\omega = 45°$ surface in figure 37 is superimposed on the plots in figure 62. One sees that the behaviors of the two heating modes are quite different. With subcooling, the surface consistently diverges to a multivalued shape, and one is unable to obtain convergence between the flowfield and surface solutions. The plot in figure 62 actually represents the steady-state solution after the 16th surface iteration. With subsequent iterations, the depression near the center-line grows until the surface becomes multivalued at $x_1 \approx 0.1$ and 0.9. An attempt was made to run the case with a finer 34 by 25 element grid, but the same instability was encountered. Initially, the surface appears to converge in the test following each steady-state solution. However, after ~16 iterations, it begins to diverge.

The shift from convergent to divergent behavior indicates that interfacial pressure is strongly influenced by the contour and growing depression in the middle of the cavity. During the first few iterations, the meniscus assumes an inflection to accommodate the negative contribution of P_d to the contour angle integration. The flat surface around this inflection extends the high pressure region near the interline into the center and drives the depression even lower. Another contributing effect is the increase in circulation intensity with each iteration pass. Figure 62 shows that the circulation for the deforming case is greater than its fixed counterpart in figure 37. The depressed surface around the center-line serves to increase the x_2-component of the thermocapillary stress force $Bi\Theta_2$, and promotes circulation. Consequently, the deformation associated with subcooling has the same stress-related effect on the interline pressure gradient as reducing contact angle.

With superheating, convergence to a stable steady-state solution occurs in only eight surface iterations. As it approaches this state, the meniscus flattens and decreases in area due to the pressure drop towards the wall. Although the deformed flowfield is very similar to the case in figure 37, one notable difference is the slightly reduced half-cavity circulation. The deformation here models

the effect of increasing contact angle and reducing $Bi\Theta_2$. This lessens the interline pressure gradient and ensures stable numerical convergence. As long as the surface retains a positive curvature, superheated thermocapillary flow should promote mechanical equilibrium along the interface.

Next, the effect of reducing Rs to 1 is examined while holding Bi and Ma constant at 10 and 10^2, respectively. This change not only increases the influence of interfacial flow on the flowfield, but it also yields a substantially larger recoil parameter value of $Vr = 1$. One sees in figure 63 that, apart from the change in position of the surface, the temperature fields do not vary much from the distributions in figure 37. As before, it is impossible to obtain a steady-state solution for subcooling. The flowfield shown here corresponds to the 16th surface iteration. However, the number of iterations required to reach the multivalued condition is greater than the case in figure 62. This suggests that recoil does provide an offsetting effect and suppresses the instability associated with dynamic pressure. With superheating, reducing the interfacial resistance causes a slight rise in the surface relative to the case in figure 62. Since pressure and recoil both provide positive contributions to curvature, the deformed surface exhibits the same stabilizing effect on pressure gradient.

Figure 63. $Ma = 10^2$, $Rs = 1$, $Bi = 10$, and $Cr = 10^{-1}$.

It appears that vapor recoil can reduce or possibly eliminate the mechanical nonequilibrium arising from subcooled thermocapillary flow. This is verified in figure 64 which shows the solutions corresponding to $Rs = 10^{-1/2} = 0.3162$ and $Vr = 10$. The subcooled case is completely stable and slightly raised with respect to the SIS curve. The order of magnitude increase in Vr is sufficient to offset the influence of dynamic pressure. With superheating, one encounters the same flattening of the meniscus as before. Because of the increased recoil, the change from the static isothermal state is much greater. Deformation also influences interfacial heat and mass transfer. In the nondeforming cases in figure 38, the superheated regimes had generally larger circulations and throughputs than their subcooled counterparts. When deformation is considered, however, the throughput and circulation intensity for subcooling is greater. The increase in throughput is attributed to the larger exposed surface area for subcooled flow, while the difference in circulation is due to the flatter surfaces and lower $Bi\Theta_2$ associated with superheating.

Figure 64. $Ma = 10^2$, $Rs = 10^{-1/2}$, $Bi = 10$, and $Cr = 10^{-1}$.

All of the subcooled cases considered so far have included the presence of thermocapillary flow and its destabilizing effect on surface morphology. In figure 65, one neglects this form of convection and examines the case of pure interfacial flow. By ignoring surface stress ($Ma = 0$), one removes the strong interline pressure gradient arising from thermocapillarity and directly assesses the influence of interfacial temperature on deformation. Comparison of the plots in figure 65 indicates that the difference in surface geometry for the two heating modes is relatively small. The slight difference reflects the manner in which interfacial convection either increases or decreases thermal potential. With subcooling, condensation of warm vapor tends to reduce Π and recoil by lowering the temperature difference between the liquid and vapor. With evaporation, however, convection from the bottom of the cavity increases the liquid/vapor temperature difference and recoil force. This stronger recoil force, in turn, leads to a more pronounced flattening of the meniscus.

The results, so far, have coincided closely with the first-order behavior investigated in section VII. One aspect that has not been addressed is the effect of surface inflections on stability. One is interested in determining whether steady-state solutions exist for situations involving a change in the sign of surface curvature.
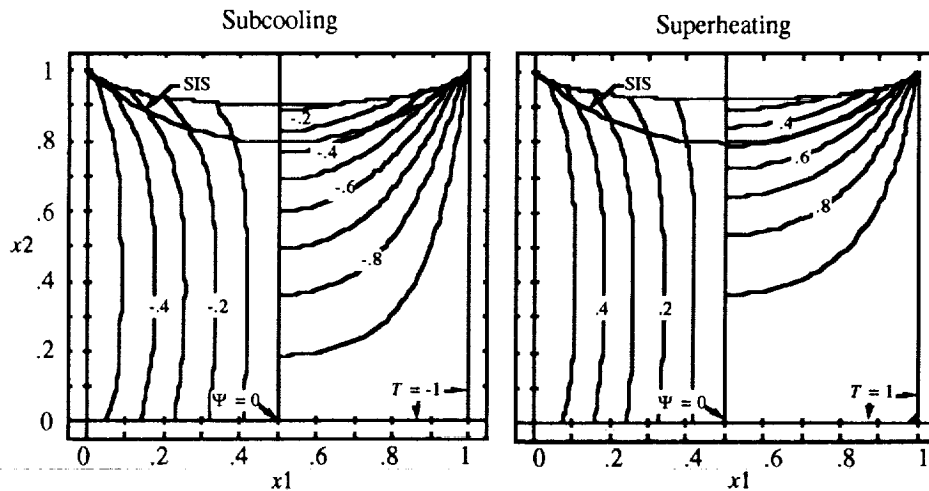
Figure 65. $Ma = 0$, $Rs = 10^{-1/2}$, $Bi = 10$, $Cr = 10^{-1}$, and $Ca = 10^{-3}$.

The pressure and recoil-related deformation resulting from superheating causes an upward-pointing bulge in the middle of the meniscus. Since the surface in figure 64 is nearly perfectly flat, it is probable that a further reduction in Rs should yield an inflection and negative curvature at the center-line. This was tested by reducing Rs from $10^{-1/2}$ to $20^{-1/2}$ and doubling Vr from 10 to 20. The resulting steady-state flowfield is shown in figure 66.
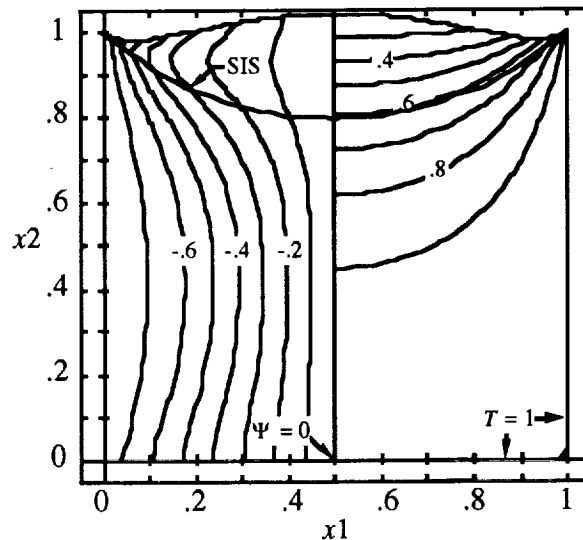
Figure 66. Superheating—$Ma = 10^2$, $Rs = 20^{-1/2}$, $Bi = 10$, and $Cr = 10^{-1}$.

Apparently, a stable solution can be obtained with a substantial negative curvature at the center-line. A unique aspect of the iteration procedure is that once the inflection is reached, recoil-related deformation serves to increase the exposed surface area, while at higher Rs, it tends to reduce it. The larger area causes an increase in flowrate, surface potential, and temperature, all of which contribute to a larger recoil force. Such a situation could be unstable if the area-increasing effect of recoil and temperature-raising influence of convection reinforces each other. As it turns out, the average temperature, which is more indicative of temperature distribution for a variable surface, actually decreases once the surface passes the inflection point. If recoil exhibits a greater dependency on this parameter, then such a situation would promote stability up to and beyond the multivalued limit.

Although convergence was achieved in the case shown in figure 66, the stability of inflections with negative curvatures is still questionable and deserves further investigation. With subcooling, one failed to obtain a stable, pressure-dominated surface with any clearly evident inflection. The closest that this report came to this condition was at $Rs = 5^{-1/2}$, where a surface that projected slightly below the SIS contour was obtained. As figure 67 shows, it is not apparent whether this surface contains an inflection. For all values of $Rs > 5^{-1/2}$, the iteration converged until the surface reached an inflection, but because no convergence was established, the iteration diverged beyond this point to the multivalued limit.

It appears that the interfacial pressure arising from thermocapillarity has a destabilizing effect on a subcooled surface. The recoil mechanism, however, tends to counteract this effect and foster mechanical equilibrium. It is probable that pressure-induced deformation is the chief cause for retention loss in LAD screens with liquid hydrogen. In these devices, pressurization with a warm vapor subcools the liquid, and through thermocapillary stress, establishes a suction that pulls the

meniscus down in the middle of the pore. Although such conditions also promote the offsetting effect of condensation-induced recoil, the Rs values corresponding to the pore size range of interest are too low to produce a force large enough to counteract the destabilizing influence of dynamic pressure.
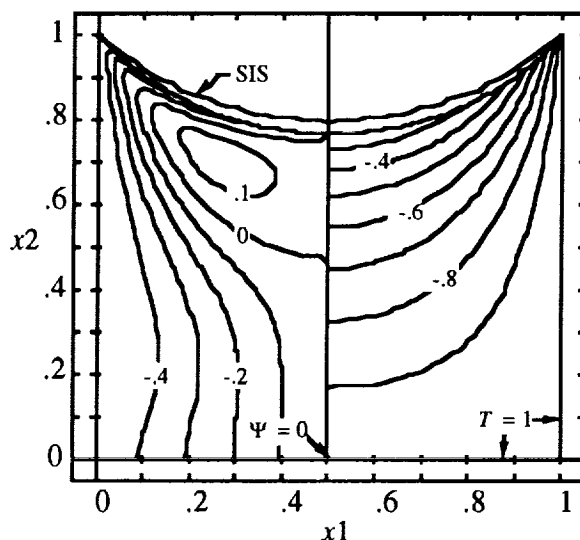
Figure 67. Subcooling—$Ma = 10^2$, $Rs = 5^{-1/2}$, $Bi = 10$, and $Cr = 10^{-1}$.

The results also explain the difference observed between screens subjected to heated pressurant and those heated directly by conduction. Heating through solid contact with the liquid models the situation of a superheated pore. One would expect the establishment of a center-directed thermocapillary flow pattern at the surface and a suction in the vicinity of the sidewalls. Unlike subcooling, this would promote stability and result in a more robust meniscus.

C. Reduction in Accommodation Coefficient

In the previous section, the effect of changing vapor/liquid equilibrium while holding Bi constant was examined. From the standpoint of liquid acquisition devices, this approach was adequate for identifying causes for retention loss in situations involving pressurization with a heated vapor. However, it is insufficient for explaining why this failure mode does not arise when the device is pressurized with an inert gas, such as helium. To examine the influence of an inert pressurant, one must account for the relationship between Bi and the level of equilibrium between the vapor and liquid (i.e., Rs). As discussed in sections III and VI, maintaining a constant Bi is equivalent to varying the evaporation number according to $E \propto Rs^{-1}$. This is physically unrealistic since E is independent of vapor/liquid equilibrium, and depends solely on liquid properties and the imposed temperature difference.

The degree of vapor/liquid equilibrium is embodied in the accommodation coefficient e in equation (27), which represents the resistance to interfacial mass transport. Its value ranges from 0 to 1, and is lowered by the presence of foreign molecules in either the condensed or vapor phase. It has previously been assumed that $e = 1$, which is appropriate for modeling kinetics of a pure liquid in contact with its own saturated vapor. According to gas kinetic theory, an inert gas should suppress interaction between the liquid and vapor, and from the standpoint of the one-sided model, decrease the effective accommodation coefficient. One can see from equation (41) that a reduction in e is

110

primarily manifested as an increase in Rs and interfacial nonequilibrium. It is also apparent from equations (57) and (48) that a lower e will yield a more convection-limited temperature distribution (i.e., lower Bi) and a decreased recoil force (i.e., lower Vr and Θ_2). For this study, the presence of an inert gas can be modeled by increasing Rs and lowering the value of Bi according to equation (57), while holding E constant.

The reduction in Bi reflects a decrease in latent heat transport between the liquid and vapor, which tends to lower the interfacial temperature gradient and thermocapillary stress along the surface. The consequence of this for subcooling is twofold. The lowering of Vr will decrease the offsetting and stabilizing effect of recoil. However, the corresponding decrease in thermocapillary stress will reduce circulation and the magnitude of the interline pressure gradient. The net result should be a reduction in pressure-induced deformation.

The influence of an inert gas and reduction in accommodation coefficient is assessed by comparing figures 63, 68, and 69. In this series of cases, E is held at 10^{-1} while Rs is raised incrementally from 1 to 2 to 10. The corresponding Biot number change is from 10 to 5 to 1. As noted before, the subcooling regime in figure 63 is unsolvable due to the destabilizing effect of dynamic pressure, while the superheating regime is stable.
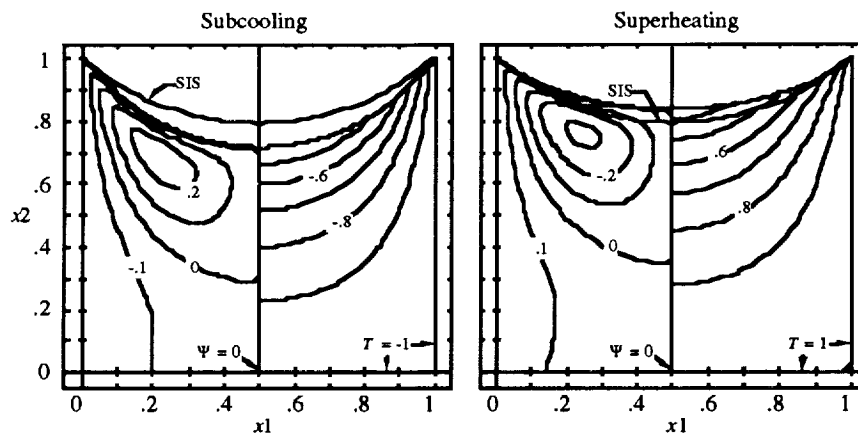
Figure 68. $Ma = 10^2$, $Rs = 2$, $Bi = 5$, and $Cr = 10^{-1}$.
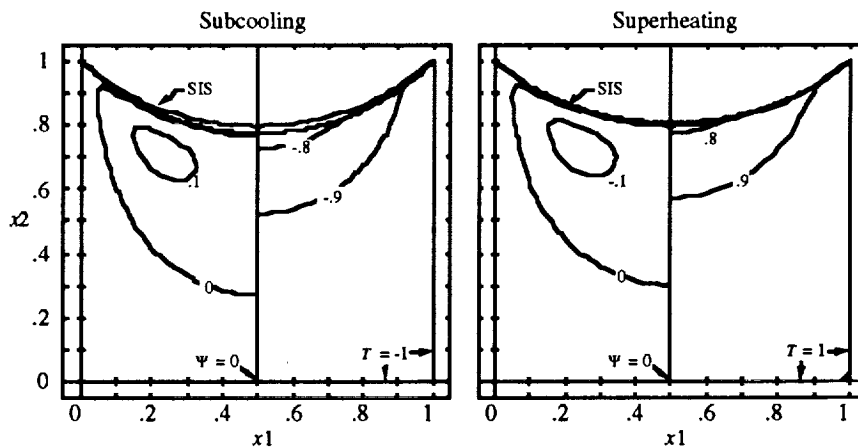
Figure 69. $Ma = 10^2$, $Rs = 10$, $Bi = 1$, and $Cr = 10^{-1}$.

111

At the next increment of Rs (= 2), a stable solution for the subcooling regime is obtained. As shown in figure 68, the steady-state flowfield solution exhibits a slight depression around the center-line. Although no inflection is clearly apparent, the surface does become relatively flat near the sidewalls. One also sees that the surface temperature gradient is lower than the case in figure 63, and causes a drop in circulation intensity and thermocapillary convection. With superheating, one notes the same general thermal behavior. Here, however, the increase in Rs and reduction in interline pressure gradient decreases the effect of recoil and dynamic pressure causing the surface to drop closer to the SIS curve.

With Rs = 10 (fig. 69), a dramatic drop in circulation intensity is noted for both heating modes. Considerable reduction in viscous losses at the sidewall eliminates the main factor causing the difference in circulation intensity between subcooling and superheating. Most importantly, the surface in both cases converges to a shape which is very close to the static isothermal geometry.

Reduction in accommodation coefficient appears to restrict both the intensity of circulation in the pore and the deformation associated with pressure and recoil-induced stress. This is especially true for subcooled thermocapillary flow, which, based on a simple decrease in e, can transition from an unstable solution to one that is completely well behaved. Although this phenomenon was only sketchily studied, it appears to provide a plausible explanation for the resistance to retention failure exhibited in LAD tests with gaseous helium. Regardless of whether the gas presents a subcooled or superheated environment, the transition into a more convection-limited interfacial temperature distribution suppresses the pressure gradients associated with thermocapillary flow.

D. Summary

In this section, the steady-state solution of pores with a deformable surface was briefly examined. Using parameter values approximately representative of liquid hydrogen exposed to its own vapor, the response of the surface to several heating modes was evaluated, including pressurization with heated vapor, screen (solid boundary) conduction, and pressurization with inert gas. This was achieved by varying the level of vapor/liquid equilibrium for both superheating and subcooling cases.

It was found that the surface was relatively immune to superheating. Pressure and recoil forces both act in the same direction and depress the meniscus near the sidewalls, thus duplicating the effect of large contact angles. This reduces the circulation associated with thermocapillarity, suppresses pressure variation along the surface, and tends to stabilize the surface. With subcooling, however, the surface was particularly sensitive to thermocapillarity and pressure-induced stress effects which cause a suction and drawing down of the meniscus in the middle of the cavity. In fact, one was unable to obtain a steady-state solution for the case of hydrogen exposed to its own vapor.

The response to an inert gas was evaluated by reducing the accommodation coefficient, which increases Rs and decreases Bi. The less severe temperature gradient restricts circulation and interfacial pressure gradient, and suppresses the mechanical nonequilibrium associated with subcooled thermocapillary flow.

IX. DISCUSSION & CONCLUSIONS

A. Introduction

The objective of this study was to determine if the convection arising from thermodynamic nonequilibrium along the porous surface of LAD's could lead to the retention loss observed in previous experiments with hydrogen liquid and vapor. The study also attempted to explain why these devices seemed immune or only slightly prone to retention failure with either direct heating of the screen surface or pressurization with heated helium gas. A major accomplishment was the definition of a physical model that approximates the conditions in previous hydrogen retention experiments. This provided a basis for evaluating appropriate heating and convection modes, and examining first-order and coupled behavior of the free surface. The results provided a plausible explanation for both the susceptibility and resistance to retention loss under different non-equilibrium conditions. In obtaining these results, there were several intermediate findings that supported the final conclusions and provided insight into this relatively new and unique problem. Many of these are relevant to systems that involve liquid/vapor phase change in microgravity or along a porous surface.

B. Discussion, Liquid Retention

Pressurization of a cryogenic vessel or direct unpressurized heating of a screen yields several possible nonequilibrium states between the vapor and liquid. Pressurization of an LAD containing liquid hydrogen with heated vapor represents a (1) subcooling of the liquid in the presence of a saturated (or superheated) vapor. Direct contact heating of the screen, however, models (2) superheating of the liquid in the presence of a saturated vapor. Introduction of an inert gas in a hydrogen pressurization system represents a (3) subcooling of the liquid in the presence of an inert gas. Finally, pressurization with helium has the same effect as (4) superheating the liquid in the presence of an inert gas.

1. <u>Subcooling with Saturated Vapor (Case 1)</u>. Previous tests showed that case 1 was most prone to retention failure. That observation is supported by the results of this study which point to thermocapillary flow arising from condensation as the cause for retention loss. When the liquid is subcooled, condensation of warm vapor raises surface temperature and establishes a negative temperature gradient towards the sides of the pore. This gradient yields a thermocapillary stress that drives flow at the surface towards the wires (i.e., side walls). With a highly wetting fluid, such as hydrogen, this flow pattern produces a large positive pressure gradient in the wire vicinity. If the meniscus is at a position corresponding to the nominal bubble point, this pressure distribution is manifested as a suction in the middle of the surface. Because of the small pore dimensions and Capillary numbers ($Ca \sim 10^{-3}$), the surface is highly sensitive to this suction and deforms into the liquid.

The geometrical limitations imposed by use of quadrilateral finite elements prevented considering a zero contact angle. Even if one could have accurately modeled such a situation, a zero contact angle would have been difficult to accommodate with a growing depression in the center of the pore. This is because the surface on either side of the center-line would have to assume a double inflection to maintain the bubble point position and satisfy the zero contact angle constraint. If the meniscus was at the nominal bubble point prior to pressurization, then, at the onset of subcooling, thermocapillary-induced circulation would cause it to detach and move into the liquid.

Although a more conservative 45° contact angle was used in the investigation, the same destabilizing effect was observed with subcooling. Here, the growing depression increases the thermocapillary stress force and half-cavity circulation, which drives the pressure gradient even higher. As discussed in section VIII, an increasing curvature at the center-line has the same effect as decreasing contact angle, and strengthens the thermocapillary force in the x_2-direction. This is physically unstable, since the surface calculated after each steady-state solution serves to increase the deformation-producing effect of dynamic pressure. The author was unable to model complete expansion of the surface into the liquid because of the limited numerical representation of interfacial temperature and pressure. However, the point of retention loss probably occurs when the surface becomes multivalued with respect to x_1. After this point, continued deformation with a constant back pressure reduces curvature and resistance to further expansion into the liquid.

Although retention loss is attributed to condensation, the recoil mechanism has little effect on this phenomenon. Because of hydrogen's low ratio between liquid and vapor density (i.e., $f_\rho \approx 50$), the recoil parameter is too low to provide significant contributions to curvature. If it was larger, recoil could offset pressure-induced deformation and improve retention performance.

2. Superheating with Saturated Vapor (Case 2). The evaporation caused by superheating liquid in the presence of pure vapor yields a positive temperature gradient in the direction of the wires and a flowfield structure opposite to that in case 1. Thermocapillary flow along the surface establishes a negative pressure gradient towards the side wall, which reduces curvature in the middle of the cavity. Because of the small recoil contribution, this deflection is primarily attributed to the pressure increase. The reduction in curvature duplicates the same effect as increasing contact angle and lowers the thermocapillary stress force. Since this tends to reduce the pressure gradient, the situation is inherently stable and exhibits no sign of retention loss.

The conclusion that superheating promotes mechanical equilibrium only applies to the surface and is valid as long as flow through the bottom of the cavity replenishes evaporative losses. In instances where wicking is unable to accommodate this rate, the surface will be susceptible to dryout and loss of retention.

3. Subcooling with Inert Gas (Case 3). The situation of subcooling in the presence of an inert gas occurs when a tank partially filled with gaseous helium is pressurized with warm hydrogen vapor. As in case 1, condensation takes place but at a rate substantially less due to the increase in nonequilibrium along the surface. The presence of helium is manifested by a reduction in the accommodation coefficient e. The most significant aspect of this reduction is a decrease in Biot number and temperature gradient along the surface. The associated lowering of thermocapillary stress and circulation results in a lower pressure gradient than case 1 and decrease in deformation. The presence of helium, therefore, promotes stability and improves retention performance.

4. Superheating with Inert Gas (Case 4). Case 4 models a similar situation in which the tank is pressurized entirely with heated helium. Here, however, the partial pressure of hydrogen in the gas is less than the vapor pressure of the liquid and one expects evaporation along the surface. The effect on accommodation coefficient, however, is very similar to case 3, and causes a lower Biot number and higher interfacial resistance than its evaporation counterpart in case 2. Provided that losses at the surface are adequately replenished (i.e., no screen dryout), this situation is also inherently stable.

114

C. Discussion, Summary of Heating and Flow Regimes

Before addressing the problem of retention loss in sections VII and VIII, several different heating and flow regimes were examined to improve understanding of the relationship between interfacial equilibrium, contact angle and flowfield behavior.

1. Basic State. Analysis of the basic state provided insight into how contact angle, surface curvature and Biot number Bi influence interfacial temperature. Thermal characteristics of a static wetting fluid in microgravity depend on Bi and contact angle ω—the latter of which dictates the surface orientation with respect to the temperature gradient. Although convection is ignored in the basic state, the author was able to characterize the fundamental influence of Bi and ω on heat transfer, mass transfer and stress by using the thermal potential Π, modified stress θ_i and Re-normalized thermocapillary stress force $Bi\Theta_i$. These parameters were derived by factoring out the linear effects of Re (or Ma) and Bi, and restricting our focus to the nonlinear influence of ω and Bi.

Applying the definition of θ_i, it was found that low-contact angle surfaces exhibit lower surface temperature gradients and stress distributions than flatter ones. Because the surface is closer and more parallel to the side wall in the contact region, the change in temperature with respect to the surface contour is less.

Integration of the modified stress distribution provided an indication of the total stress force acting in each direction Θ_i. The product of Bi and Θ_i reflected the sensitivity of surface traction to Bi and ω. With a wetting surface, it was found that a net thermocapillary force arises in the $+x_2$-direction with subcooling and $-x_2$-direction with superheating. No net force occurs in the x_1-direction due to the mirror-symmetry of θ_1 about the center-line. Because of the steeper surface orientation near the contact line, the magnitude of the thermocapillary force is greater for smaller contact angles.

Applying the definition of thermal potential Π, it was found that heat and mass transfer should increase at higher Bi and lower ω. Although the nonlinear effects of Bi tend to decrease Π by reducing interfacial temperature, the linear influence of Bi offsets this and increases interfacial transport. Also, the lowering of contact angle tends to raise Π due to the increase in exposed area near the side wall and along the meniscus in general.

2. Pure Evaporation and Condensation. Evaporation occurs when the boundaries are superheated with respect to the vapor, while condensation arises when they are subcooled. Evaporation establishes a vertically oriented flowfield in the $+x_2$-direction which produces counterclockwise and clockwise circulation in the left and right half-cavities, respectively. Condensation produces a similar flow structure that is opposite in direction and sign. In both cases, there is a small contribution of surface velocity to circulation (i.e., side wall bias) due to the concentration of mass flux at the side walls.

Depending on the level of vapor/liquid equilibrium, this vertically oriented flowfield can strongly influence the interfacial temperature distribution via convection and complicate the calculation of heat and mass transfer through the cavity. Evaporation convects superheated liquid to the surface which tends to raise surface temperature. Condensation causes the accumulation of warm liquid on the surface which also increases surface temperature. The temperature increase associated with evaporation, however, serves to raise thermal potential and interfacial transport, while with condensation, the potential is lowered, thus suppressing heat and mass transfer. The deviation in thermal potential from the basic state is maximized when $1 < Bi < 10$.

Evaporation and. condensation also exhibit different effects on the surface temperature gradient. The increase in surface temperature that accompanies condensation serves to raise the average gradient and extend the variation over a larger portion of the cavity. With evaporation, however, the elevated surface temperature decreases the gradient and shifts most of the temperature change to the side walls. Although the shift in gradient caused by convection has little influence on pure evaporation or condensation, it does affect thermocapillary stress with combined-mode convection.

At the same levels of surface equilibrium and superheat/subcooling, the magnitude of half-cavity circulation for evaporation is typically greater than that for condensation. This is because for $Bi < 10^2$, the primary contribution to circulation comes from the velocity along the center-line. The center-line velocities are greater for evaporation because of its reinforcing effect on thermal potential and mass transfer.

3. **Pure Thermocapillary Flow**. Thermocapillary-induced stress establishes two counter-rotating cells symmetric about the cavity center-line. With superheating, the circulation in the left half-cavity is clockwise, and warm liquid at the surface is convected from the side walls to the center-line. With subcooling the circulation is reversed, and surface fluid flows towards the side walls. Surface velocity and circulation strength depend on Ma and the variation of temperature gradient along the meniscus, which is primarily dictated by Bi. Note that the thermocapillary and interfacial circulations concomitant with either superheating or subcooling are opposite in sign.

The surface convection associated with superheating tends to increase interfacial temperature and thermal potential. Fluid in the upper portion of the cell enters the surface region near the maximum cavity temperature and flows to a cooler region. Hence, the temperature is minimum in the center of the cavity. With subcooling, the situation is more complex. Competing with the accumulation of warm liquid at the meniscus center is the return flow of cool liquid from the bottom of the cavity. This upwelling suppresses the temperature rise in the center-line region, and causes the temperature profile to exhibit a maximum between the center-line and side wall. It also offsets the convection of warm liquid towards the side walls and yields a more modest increase in thermal potential compared to superheating.

The half-cavity circulation for superheating is consistently greater than that for subcooling. At low Ma, the difference between the two is maximized at $Bi \sim 10$, but disappears as $Bi \to 0$ or $Bi \to \infty$. At high Ma, the effect of convection on the superheated temperature field is more extensive and forms a high gradient region around the center-line. There is consequently a significant difference between these two modes over the entire range of Bi.

Thermocapillary flow yields substantial pressure gradients near the side walls. With superheating, the gradient is negative because the flow towards the center-line produces a suction in the contact region. With subcooling, the flow towards the side wall produces a suction at the center-line and a pressure rise at the corners. At low Ma, the pressure rise for subcooling is larger, since the differences in circulation are rather small, and the magnitude of thermocapillary stress for subcooling is greater at the corners. At higher Ma, the substantially higher circulation for superheating yields a larger change in pressure along the surface.

The circulation and pressure variation associated with thermocapillary flow strongly depend on contact angle. With large contact angles ($\omega \to 90°$), the circulation in each half-cavity is suppressed by opposing flow from the other side. At lower contact angles, the circulation for both

116

heating modes is more intense. It was also found that the circulation intensity is directly related to the x_2-component of the net thermocapillary stress force exerted on the surface. The strong dependence of circulation on contact angle also applies to the behavior of dynamic pressure. At high angles, the pressure distribution is somewhat constant, which reflects the reduced constriction on flow area at the interline. At low angles, however, the pressure gradient at the side walls is substantially greater.

4. Combined-Mode Convection. In general, the behavior of pure thermocapillary and interfacial flow still applies in the combined situation, even when one mode clearly dominates over the other. There are, however, several unique features of this type of regime. For one, the circulations associated with thermocapillary and interfacial flow oppose each other. Thus, when both modes are present, the total half-cavity circulation C is a good indicator of which mode governs the convection pattern and temperature distribution. Although the circulations are opposite in sign, their simultaneous presence may actually reinforce the effects of both modes. For instance, with subcooling in the transition region ($C \approx 0$) thermocapillarity tends to augment the suppression of thermal potential caused by condensation. By distending the warm temperature region towards the side wall, the thermal potential decreases to a value lower than that for pure condensation.

The pressure gradient for combined flow is also greater than either its pure thermocapillary or interfacial flow counterpart. This is because combined flow reflects a superposition of the pressure distributions for thermocapillary and interfacial flow.

5. Surface Response and Morphology. The thermal dependence of surface tension appears to have little effect on surface curvature and steady-state morphology. The first-order analysis in section VII showed that this term would be significant only if the Crispation number Cr were close to 1. This is physically unrealistic, since Cr is usually less than 10^{-1}.

The surface is most sensitive to viscous stress in instances where there is strong thermocapillary flow along a highly curved meniscus. The curvature term in the expression for viscous stress appears to exhibit a much stronger sensitivity than the flux component, especially at small contact angles. However, compared to the other terms in the normal stress balance, viscous stress can be safely ignored for the problem considered here.

For the length and temperature scales considered in this problem, the two most important terms in the normal stress equation are vapor recoil and dynamic pressure. The surface deformation associated with recoil is the same for both superheating and subcooling and causes a flattening and, under extreme conditions, a bulging in the center of the pore. It appears, however, that this deformation is stable in that the response does not result in an increase in the temperature distribution across the meniscus. In fact, its effect is analogous to increasing the contact angle.

The influence of dynamic pressure represents the only potential instability. With superheating, this term has the same effect as recoil and tends to flatten the meniscus. It duplicates the same effect as increasing contact angle and serves to reduce the magnitude of pressure gradient at the side walls. The deformation in this case is stable. With subcooling, however, the deformation is reversed and causes a depression about the center-line. This promotes circulation and tends to increase the magnitude of the pressure gradient and the deformation even further. The effect of pressure in this case is analogous to decreasing contact angle. It appears that pressure-induced deformation primarily depends on the circulation caused by thermocapillary stress and becomes more pronounced by increasing either Ma or Bi.

D. Conclusions

The results of this study support the original hypothesis of retention loss being caused by convection within the screen pores of liquid acquisition devices. The retention failures encountered in previous experiments with liquid hydrogen and pressurized heated hydrogen vapor are caused by the thermocapillary flow arising from condensation-induced temperature gradients along the liquid surface of each pore. This flow establishes an interfacial pressure variation that deforms the center of the surface into the liquid. The situation is physically unstable at zero to low contact angles, since the deformation serves to increase the net thermocapillary stress force and pressure gradient even further. The end result of the deformation is an expansion of the meniscus into the liquid and eventual detachment from the screen wires.

The resistance to retention loss observed with heated helium pressurant is due to the change in direction of heat and mass transfer at the surface. In this case, the surface evaporates and establishes a thermocapillary flow structure that is opposite to the case of hydrogen pressurization. Instead of depressing the meniscus, the pressure gradient tends to raise the surface in the center of the pore. This situation is inherently stable because deformation in this direction serves to reduce the thermocapillary stress force, circulation and interfacial pressure gradient. As long as the wicking rate of the screen can accommodate evaporative losses, pressurization with heated helium promotes mechanical equilibrium and mitigates retention loss.

The immunity to retention loss encountered with direct heating of screen samples is due to the same behavior associated with helium pressurization. Since the vapor surrounding the screen is not pressurized relative to the liquid, the pore menisci are superheated relative to the vapor. The same thermocapillary flow pattern encountered with helium pressurization develops, and, provided the wicking rate is adequate, the screen exhibits nominal or improved retention characteristics.

E. Recommendations

1. Liquid Acquisition Device Applications. From a spacecraft design standpoint, this study has furthered understanding of a second retention failure mode for screened LAD's. The first, which has been recognized for years, is the dryout caused by evaporation and inadequate wicking through the screen structure. This problem, however, is correctable with proper design of the liquid and screen system. Most importantly, the failure is not attributable to a liquid surface effect, since, as this study showed, evaporation tends to promote surface stability and does not detract from meniscus retention behavior.

The failure mode investigated in this study is completely different than screen dryout. It is caused by the presence of a physically unstable flow pattern in the pores of the screen and occurs only when heated vapor is used as the pressurization source. This retention loss mechanism would present problems only in cases of autogenous pressurization.

The results of this study suggest that an autogenous system is incompatible for use with liquid hydrogen. Thus, the most straightforward recommendation is to discourage use of such a system. This study has confirmed that the traditional approach of using an independent pressurant source, such as helium, has little impact on retention performance. However, autogenous pressurization is a recent concept that represents an advancement over helium pressurization, and is appealing due to its potential reduction in weight and complexity.

118

Based on the results of this study, it appears that loss of retention in an autogenous system could be mitigated by either (1) reducing the accommodation coefficient and interfacial thermal gradients in the pore menisci, (2) reducing the level of subcooling between the vapor pressurant and liquid near the LAD surface, or (3) eliminating the pore liquid surface and thermocapillary convection altogether during periods of pressurization.

There are several ways of reducing accommodation coefficient. One that is complementary with some current pressurization schemes is to inject and mix a small amount of helium with the hydrogen vapor. The presence of helium reduces the accommodation coefficient and increases non-equilibrium at the surface. This suppresses condensation, reduces the interfacial temperature gradient, and ultimately reduces thermocapillary flow and the flow-induced pressure gradient responsible for retention loss.

Another method, which is appealing for start basket applications, is to employ helium only during the initial phases of pressurization. Retention loss usually occurs prior to engine restart when the outside screen surface is exposed to vapor. During this period, a small independent supply of helium is used to ensure engine restart. Once the liquid reorients and fills the basket, the autogenous supply is applied for the duration of the maneuver. The amount of helium required is much less than a totally independent system. However, if the mass diffusion rate of helium in hydrogen is much less than thermal diffusion in the tank, then this concept incurs the same retention difficulties as an autogenous system.

Another alternative for reducing accommodation coefficient is to add a foreign substance to the hydrogen liquid itself. Seeding of hydrogen has been considered before for other applications, such as magnetic control. However, the issue always arises on how the substance will impact propellant properties and rocket engine performance.

A second method for correcting retention failure is to offset subcooling by heating the pore directly. This approach may be viable if the heat could be applied uniformly across the screen surface. However, the complex geometry of the screen and inefficiencies associated with conduction along such a tortuous path requires application of a relatively large heat load at only a few locations. Not only would the level of heating and offset to subcooling vary considerably along the screen surface, but in the vicinity of heat application, the liquid could become highly superheated and susceptible to retention loss via evaporation and dryout.

The third method of mitigating retention loss is to devise a way of eliminating the porous surface entirely. Recall that the problem arose from condensation-induced thermocapillary convection on the liquid surfaces of individual pores. If, on the other hand, condensation occurred along not only the liquid surfaces but also the screen wires, then this convection mode would be altered and the adverse pressure gradients caused by flow in the contact region would be eliminated. Key to this approach is the establishment of a liquid layer prior to development of a thermocapillary flowfield. This concept holds promise since the subcooling could be provided relatively easily by the Joule-Thompson effect in a thermodynamic vent system.

2. Future Research. In addition to recommendations pertaining to the design of LAD's and autogenous pressurization systems, several suggestions are offered for future research and investigations in this area. Most of these are in response to issues raised in this study and include:

119

a. The problem domain considered here was very simple and did not realistically portray the situation near an LAD screen wire. Future investigations could concentrate on more complex geometries, such as the type in figure 3, or even problems involving solution of a three-dimensional flowfield. This would more accurately reflect the situation in a liquid acquisition device, heat pipe, or other porous engineering systems.

b. The isothermal boundary conditions for the solid and internal interface did not reflect the wider range of operating conditions encountered in actual applications. Future investigations could examine the effect of using a Neumann-type heat flux condition at the side wall and possibly a liquid in a saturated state coming from the interior. This would more properly model the situation of applied heating at the interface.

c. The analysis of surface deformation and stability was rather limited in terms of sophistication and breadth. The purpose here was merely to identify probable retention failure mechanisms. Characterizing the nature and details of this behavior, particularly the destabilizing effect of dynamic pressure, should be investigated using more sophisticated stability analyses.

d. Contact angle appears to play a major role in thermocapillary circulation and should be examined more thoroughly.

F. Summary

In this section, the findings and results have been summarized in a manner that relates to both LAD retention and two-phase microgravity fluid convection. The conclusions have been presented which provide a plausible explanation for the disparity in liquid hydrogen retention observed with different pressurant and heating conditions. To make the results more relevant from an engineering standpoint, several recommendations have been offered for mitigating and eliminating this form of retention loss in autogenous pressurization systems. Although all of these have potential drawbacks, it is important that condensation-induced retention loss be recognized as an additional consideration in the design of LAD's and spacecraft propellant systems.

APPENDIX A

DOMAIN ELEMENT INTERPOLATION FUNCTIONS

A. Introduction

Several different types of interpolation functions Φ_N are used to approximate the variation of velocity, pressure, and temperature on the element level. The form of these functions are distinguished by their order of variance with respect to the element naturalized coordinates ξ_1 and ξ_2 (or ξ and η, respectively). A diagram illustrating the relationship between a representative quadrilateral element and the local coordinate frame is shown in figure A-1.
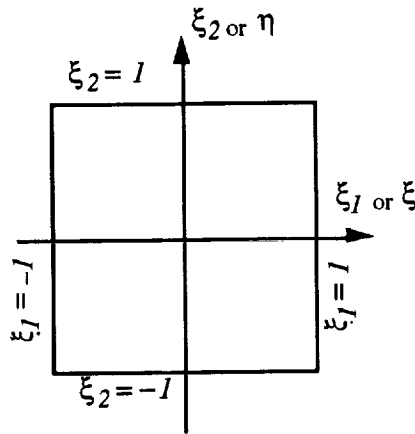
Figure A-1. Naturalized coordinate frame.

The index N of the base function Φ_N refers to the local node at which the function applies. The derivation of expressions for Φ_N is based on the nodal positions designated in figure 6. These designations for node positions are used consistently throughout appendix A.

As was discussed in section IV, the simplest elements employ the same order of variance in both the ξ_1 and ξ_2 directions (i.e., types 1, 3, and 5). This form is used for all the interior elements shown in figure 6, and consists of first-, second-, and fourth-order polynomials expressed in terms of ξ_1 and ξ_2. The other type of elements employ different interpolation orders in each direction (i.e., types 2 and 5). Along the meniscus, pressure is modeled by a first-order interpolation in the ξ_1-direction and a second-order in the ξ_2-direction, while temperature is approximated as a second-order in the ξ_1-direction and fourth-order in the ξ_2-direction. For elements adjacent to the sidewall, the higher order variation is applied in the ξ_1-direction. The following chapter outlines derivation of expressions for Φ_N, and its associated first and second-order gradients with respect to the global coordinate frame x_i.

B. Lagrange Polynomials

The two-dimensional base functions Φ_N are derived by taking the product of the one-dimensional Lagrange polynomials corresponding to each direction. Each polynomial characterizes the variance in either the ξ_1 or ξ_2-direction at each element node. All of the interpolation functions

used in this study are based on and require the definition of first-, second-, and fourth-order Lagrange polynomial equations. These are obtained from the general form:

$$L(i)_N^n = \prod_{m=1/m \neq N}^{n+1} \frac{\xi_i - \xi_{mi}}{\xi_{Ni} - \xi_{mi}} \ . \tag{A-1}$$

n represents the order of the approximation, while $n+1$ represents the total number of nodes required in the ξ_i direction. ξ_{Ni} is the position of node N in terms of naturalized coordinates. For a linear variance between two endpoints, n is set equal to 1 and the products are calculated from $m = 1$ to $m = 2$. This yields the two Lagrange interpolation functions:

$$L(i)_1^I = \left(\frac{\xi_i - \xi_{2i}}{\xi_{1i} - \xi_{2i}}\right) = \left(\frac{\xi_i - 1}{-1 - 1}\right) = -\frac{1}{2}(\xi_i - 1) \ , \tag{A-2}$$

$$L(i)_2^I = \left(\frac{\xi_i - \xi_{1i}}{\xi_{2i} - \xi_{1i}}\right) = \left(\frac{\xi_i + 1}{1 + 1}\right) = -\frac{1}{2}(\xi_i + 1) \ .$$

For a second-order variance, $n = 2$, and one obtains polynomial expressions at three nodes:

$$L(i)_1^{II} = \left(\frac{\xi_i - \xi_{2i}}{\xi_{1i} - \xi_{2i}}\right)\left(\frac{\xi_i - \xi_{3i}}{\xi_{1i} - \xi_{3i}}\right) = \left(\frac{\xi_i - 0}{-1 - 0}\right)\left(\frac{\xi_i - 1}{-1 - 1}\right) = \frac{1}{2}\xi_i(\xi_i - 1) \ , \tag{A-3}$$

$$L(i)_2^{II} = \left(\frac{\xi_i - \xi_{1i}}{\xi_{2i} - \xi_{1i}}\right)\left(\frac{\xi_i - \xi_{3i}}{\xi_{2i} - \xi_{3i}}\right) = \left(\frac{\xi_i + 1}{1}\right)\left(\frac{\xi_i - 1}{-1}\right) = 1 - \xi_i^2 \ ,$$

$$L(i)_3^{II} = \left(\frac{\xi_i - \xi_{1i}}{\xi_{3i} - \xi_{1i}}\right)\left(\frac{\xi_i - \xi_{2i}}{\xi_{3i} - \xi_{2i}}\right) = \left(\frac{\xi_i + 1}{2}\right)\left(\frac{\xi_i - 1}{1}\right) = \frac{1}{2}\xi_i(\xi_i - 1) \ .$$

Third-order interpolations were considered early in the study but were not included in the final version of the model. In any event, the Lagrange polynomials associated with a third-order variation are obtained in a similar manner by setting $n = 3$.

$$L(i)_1^{III} = -\frac{9}{16}(\xi_i^2 - 1/9)(\xi_i - 1) \ , \tag{A-4}$$

$$L(i)_2^{III} = \frac{27}{16}(\xi_i^2 - 1)(\xi_i - 1/3) \ ,$$

$$L(i)_3^{III} = -\frac{27}{16}(\xi_i^2 - 1)(\xi_i + 1/3) \ ,$$

$$L(i)_4^{III} = \frac{9}{16}(\xi_i^2 - 1/9)(\xi_i + 1) \ .$$

For a fourth-order variation, $n = 4$ and one must consider polynomials evaluated at the origin, two endpoints, and two intermediate locations for a total of five nodes.

122

$$L(i)_1^{IV} = \frac{2}{3}\,\xi_i(\xi_i+1/2)\,(\xi_i-1/2)\,(\xi_i-1) \ ,$$

(A-5)

$$L(i)_2^{IV} = -\frac{8}{3}\,\xi_i(\xi_i-1/2)\,(\xi_i+1)\,(\xi_i-1) \ ,$$

$$L(i)_3^{IV} = 4(\xi_i+1/2)\,(\xi_i-1/2)\,(\xi_i+1)\,(\xi_i-1) \ ,$$

$$L(i)_4^{IV} = -\frac{8}{3}\,\xi_i(\xi_i+1/2)\,(\xi_i+1)\,(\xi_i-1) \ ,$$

$$L(i)_5^{IV} = \frac{2}{3}\,\xi_i(\xi_i+1/2)\,(\xi_i-1/2)\,(\xi_i+1) \ .$$

C. Interpolation Functions

1. Function Type 1

The two-dimensional interpolation functions with equivalent orders of approximation in the ξ_1 (or ξ) and ξ_2 (or η) directions are constructed by taking the product of the Lagrange polynomials corresponding to each local node's ξ_1 and ξ_2 coordinates.

$$\Phi_N^n = L(1)_{\xi(N)}^n L(2)_{\eta(N)}^n \ .$$

(A-6)

N signifies the element node number, while $\xi(N)$ and $\eta(N)$ are its associated naturalized coordinates. For linear interpolations, the local interpolation function that applies over the two-dimensional element domain is obtained by multiplying the ξ_1 and ξ_2 Lagrange functions for each local node.

$$\Phi_1^{II} = L(1)_1^{II} L(2)_1^{II} = \frac{1}{4}\,\xi_1\xi_2(\xi_1+1)(-\xi_2+1) \ ,$$

$$\Phi_2^I = L(1)_2^I L(2)_1^I = \frac{1}{4}\,(\xi_1+1)(-\xi_2+1) \ ,$$

$$\Phi_3^I = L(1)_2^I L(2)_2^I = \frac{1}{4}\,(\xi_1+1)(\xi_2+1) \ ,$$

$$\Phi_4^I = L(1)_1^I L(2)_2^I = \frac{1}{4}\,(-\xi_1+1)(\xi_2+1) \ .$$

These expressions can be written as a single general equation by applying the coordinate values at the node under consideration, that is ξ_{N1} and ξ_{N2}. This approach is appealing because the node values are independent of ξ_i. The general expression for Φ_N^I is:

$$\Phi_N^I = \frac{1}{4}\,(\xi_{N1}\xi_1+1)(\xi_{N2}\xi_2+1) \ .$$

(A-7)

2. Function Type 3

The same approach is applied in deriving expressions for the quadratic interpolation functions. Here one must consider three nodes in each direction and a total of nine interpolation functions. For the corner nodes ($N = 1, 3, 5,$ and 7), one has:

$$\Phi_1^{II} = L(1)_1^{II} L(2)_1^{II} = \frac{1}{4} \xi_1 \xi_2 (\xi_1 - 1)(\xi_2 - 1) \; ,$$

$$\Phi_3^{II} = L(1)_3^{II} L(2)_1^{II} = \frac{1}{4} \xi_1 \xi_2 (\xi_1 + 1)(\xi_2 - 1) \; ,$$

$$\Phi_5^{II} = L(1)_3^{II} L(2)_3^{II} = \frac{1}{4} \xi_1 \xi_2 (\xi_1 + 1)(\xi_2 + 1) \; ,$$

$$\Phi_7^{II} = L(1)_1^{II} L(2)_3^{II} = \frac{1}{4} \xi_1 \xi_2 (\xi_1 + 1)(\xi_2 + 1) \; ,$$

which can be expressed more generally as:

$$\Phi_N^{II} = \frac{1}{4} \xi_{N1} \xi_{N2} \xi_1 \xi_2 (\xi_{N1} \xi_1 + 1)(\xi_{N2} \xi_2 + 1) \; . \quad N = 1, 3, 5, \text{ and } 7 \quad \text{(A-8a)}$$

For the midside nodes (i.e., noncenter axes nodes) one obtains:

$$\Phi_2^{II} = L(1)_2^{II} L(2)_1^{II} = \frac{1}{2} \xi_2 (\xi_2 - 1)(1 - \xi_1^2) \; ,$$

$$\Phi_4^{II} = L(1)_3^{II} L(2)_2^{II} = \frac{1}{2} \xi_1 (\xi_1 + 1)(1 - \xi_2^2) \; ,$$

$$\Phi_6^{II} = L(1)_2^{II} L(2)_3^{II} = \frac{1}{2} \xi_2 (\xi_2 + 1)(1 - \xi_1^2) \; ,$$

$$\Phi_8^{II} = L(1)_1^{II} L(2)_2^{II} = \frac{1}{2} \xi_1 (\xi_1 - 1)(1 - \xi_2^2) \; ,$$

which in general terms become:

$$\Phi_N^{II} = \frac{1}{2} \xi_{N2} \xi_2 (\xi_{N2} \xi_2 + 1)(1 - \xi_1^2) \; , \qquad N = 2 \text{ and } 6 \qquad \text{(A-8b)}$$

$$\Phi_N^{II} = \frac{1}{2} \xi_{N1} \xi_1 (\xi_{N1} \xi_1 + 1)(1 - \xi_2^2) \; . \qquad N = 4 \text{ and } 8$$

Finally, for the center node one obtains:

$$\Phi_9^{II} = L(1)_2^{II} L(2)_2^{II} = (1 - \xi_1^2)(1 - \xi_2^2) \; . \qquad N = 9 \qquad \text{(A-8c)}$$

3. Function Type 5

Derivation of the fourth-order interpolation function is also based on equation (A-6), but requires considerably more algebra. The following condensed form is obtained for the local nodes lying on the outside corners, the outside edge off the coordinate axes and the inside corners of the 25-node element shown in figure 6:

$$\Phi_N^{IV} = C\big[\xi_1(\xi_1 + \xi_{N1})(4\xi_{N1}^2\xi_1^2 - 1)\big]\big[\xi_2(\xi_2 + \xi_{N2})(4\xi_{N2}^2\xi_2^2 - 1)\big] , \tag{A-9a}$$

where:

$$C = 1/36 , \qquad\qquad N = 1, 3, 7, \text{ and } 9$$

$$C = -8/18 , \qquad\qquad N = 2, 4, 6, 8, 10, 12, 14, \text{ and } 16$$

$$C = 64/9 . \qquad\qquad N = 17, 19, 21, \text{ and } 23$$

For the center node ($N = 25$) one has:

$$\Phi_N^{IV} = 16\big[(\xi_1^2 - 1)(\xi_1^2 - 1/4)\big]\big[(\xi_2^2 - 1)(\xi_2^2 - 1/4)\big] . \tag{A-9b}$$

All remaining nodes can be expressed in the general form:

$$\Phi_N^{IV} = C\big[(\xi_\alpha^2 - 1)(\xi_\alpha^2 - 1/4)\big]\big[\xi_\beta(\xi_\beta + \xi_{N\beta})(4\xi_{N\beta}^2\xi_\beta^2 - 1)\big] , \tag{A-9c}$$

where:

$$C = 4/6, \alpha = 1 \text{ and } \beta = 2 , \qquad\qquad N = 3 \text{ and } 11$$

$$C = 4/6, \alpha = 2 \text{ and } \beta = 1 , \qquad\qquad N = 7 \text{ and } 15$$

$$C = -32/3, \alpha = 1 \text{ and } \beta = 2 , \qquad\qquad N = 18 \text{ and } 22$$

$$C = -32/3, \alpha = 2 \text{ and } \beta = 1 . \qquad\qquad N = 20 \text{ and } 24$$

4. Function Type 2

The general expression for the mixed ordered functions is very similar to equation (A-6). The only difference is that it is comprised of Lagrange polynomials of different order at each local node. The general formula is:

$$\Phi_N^{n-m} = L(1)_{\xi(N)}^n L(2)_{\eta(N)}^m , \tag{A-10}$$

where n and m correspond to the orders applied in the ξ- and η-directions, respectively. The combination of first and second order interpolations are needed to maintain consistency between the number of equations and nodes. There are two types of functions which are applied at the meniscus and sidewall, namely:

$$\Phi_N^{I-II} = L(1)_{\xi(N)}^{I} L(2)_{\eta(N)}^{II} \; ,$$

$$\Phi_N^{II-I} = L(1)_{\xi(N)}^{II} L(2)_{\eta(N)}^{II} \; .$$

For the function adjacent to the meniscus, Φ_N^{I-II}, the following general equation applies:

$$\Phi_N^{I-II} = \frac{1}{4} \xi_2 (\xi_2 + \xi_{N2})(\xi_{N1}\xi_1 + 1) \; , \qquad N = 1, 2, 4, \text{ and } 5 \qquad \text{(A-11)}$$

$$\Phi_N^{I-II} = \frac{1}{2} (1 - \xi_2^2)(\xi_{N1}\xi_1 + 1) \; . \qquad N = 3 \text{ and } 6$$

For the sidewall function, Φ_N^{II-I}, one has:

$$\Phi_N^{II-I} = \frac{1}{4} \xi_1 (\xi_1 + \xi_{N1})(\xi_{N2}\xi_2 + 1) \; , \qquad N = 1, 3, 4, \text{ and } 6 \qquad \text{(A-12)}$$

$$\Phi_N^{II-I} = \frac{1}{2} (1 - \xi_1^2)(\xi_{N2}\xi_2 + 1) \; . \qquad N = 2 \text{ and } 5$$

5. Function Type 4

One applies the same approach with the combined second and fourth-order elements by again considering two types of interpolation functions:

$$\Phi_N^{II-IV} = L(1)_{\xi(N)}^{II} L(2)_{\eta(N)}^{IV} \; ,$$

$$\Phi_N^{IV-II} = L(1)_{\xi(N)}^{IV} L(2)_{\eta(N)}^{II} \; .$$

The general forms of these two are the same as in equations (A-11) and (A-12). For the meniscus function, Φ_N^{II-IV}, one has the following different forms of interpolation:

$$\Phi_N^{II-IV} = C(4\xi_{N2}^2\xi_2^2 - 1)(\xi_2^2 + \xi_{N2}\xi_2)(\xi_1^2 + \xi_{N1}\xi_1) \; , \qquad \begin{array}{l} N = 1, 3, 7, \text{ and } 9: C = 1/12 \\ N = 12, 4, 6, \text{ and } 10: C = 8/6 \end{array} \qquad \text{(A-13a)}$$

$$\Phi_N^{II-IV} = 2(\xi_2^2 - 1)(\xi_2^2 - 1/4)(\xi_1^2 + \xi_{N1}\xi_1) \; , \qquad N = 5 \text{ and } 11 \qquad \text{(A-13b)}$$

$$\Phi_N^{II-IV} = C(4\xi_{N2}^2\xi_2^2 - 1)(\xi_2^2 + \xi_{N2}\xi_2)(1 - \xi_1^2) \; , \qquad \begin{array}{l} N = 2 \text{ and } 8: C = 1/6 \\ N = 13 \text{ and } 15: C = 8/3 \end{array} \qquad \text{(A-13c)}$$

$$\Phi_N^{II-IV} = 4(\xi_2^2 - 1)(\xi_2^2 - 1/4)(1 - \xi_1^2) \; . \qquad N = 14 \qquad \text{(A-13d)}$$

For the sidewall function, Φ_N^{IV-II}, one has:

$$\Phi_N^{IV-II} = C(4\xi_{N1}^2\xi_1^2 - 1)(\xi_1^2 + \xi_{N1}\xi_1)(\xi_2^2 + \xi_{N2}\xi_2) \; , \qquad \begin{array}{l} N = 1, 5, 7, \text{ and } 11: C = 1/12 \\ N = 2, 4, 8, \text{ and } 10: C = 8/6 \end{array} \qquad \text{(A-14a)}$$

$$\Phi_N^{IV-II} = 2(\xi_1^2-1)(\xi_1^2-1/4)(\xi_2^2+\xi_{N2}\xi_2) \ , \qquad N = 3 \text{ and } 9 \qquad\qquad (A\text{-}14b)$$

$$\Phi_N^{IV-II} = C(4\xi_{N1}^2\xi_1^2-1)(\xi_1^2+\xi_{N1}\xi_1)(1-\xi_2^2) \ , \qquad \begin{array}{l} N = 6 \text{ and } 12: C = 1/6 \\ N = 13 \text{ and } 15: C = 8/3 \end{array} \qquad (A\text{-}14c)$$

$$\Phi_N^{IV-II} = 4(\xi_1^2-1)(\xi_1^2-1/4)(1-\xi_2^2) \ . \qquad N = 14 \qquad\qquad (A\text{-}14d)$$

D. First-Order Gradients

1. General Form

To derive an expression for the gradient of Φ_N with respect to x_i (i.e., $\partial\Phi_N/\partial x_i$), one recognizes that the interpolation function is expressed as a function of local or naturalized coordinates, $\Phi_N = \Phi_N(\xi_k)$, which, in turn, are functions of the global domain, $\xi_k = \xi_k(x_i)$. The gradient of Φ_N with respect to natural coordinates, therefore, is related to the global domain by means of the Jacobian, J_{ki}, where:

$$\frac{\partial\Phi_N}{\partial\xi_k} = J_{ki}\frac{\partial\Phi_N}{\partial x_i} \ , \qquad\qquad (A\text{-}15)$$

and

$$J_{ki} = \frac{\partial x_i}{\partial\xi_k} = \begin{bmatrix} \dfrac{\partial x_1}{\partial\xi_1} & \dfrac{\partial x_2}{\partial\xi_1} \\[2ex] \dfrac{\partial x_1}{\partial\xi_2} & \dfrac{\partial x_2}{\partial\xi_2} \end{bmatrix} \ .$$

It is also recognized that x_i can be expressed as a function of ξ_l by employing an alternative transformation based on J'_{il}, namely:

$$\frac{\partial\Phi_N}{\partial x_i} = J'_{il}\frac{\partial\Phi_N}{\partial\xi_l} \ . \qquad\qquad (A\text{-}16)$$

Substituting equation (A-16) into equation (A-15) yields the identity $J_{ki}J'_{il} = \delta_{kl}$, which implies that J'_{il} is equivalent to the inverse of the Jacobian matrix, i.e., $J'_{il} = (J_{il})^{-1}$. This inverse is used to express the global gradient in terms of a locally based derivative:

$$\frac{\partial\Phi_N}{\partial x_i} = (J_{il})^{-1}\frac{\partial\Phi_N}{\partial\xi_k} \ . \qquad\qquad (A\text{-}17)$$

An expression for the inverse is derived as follows:

$$(J_{ik})^{-1} = \frac{\partial\xi_k}{\partial x_i} = \frac{\{cofac(J_{ik})\}^T}{|J|} \ ,$$

$$\{cofac(J_{ik})\}^T = \begin{bmatrix} \dfrac{\partial x_2}{\partial \xi_2} & -\dfrac{\partial x_2}{\partial \xi_1} \\ -\dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_1}{\partial \xi_1} \end{bmatrix} = \dfrac{\partial x_2}{\partial \xi_2}\delta_{i1}\delta_{k1} - \dfrac{\partial x_2}{\partial \xi_1}\delta_{i1}\delta_{k2} + \dfrac{\partial x_1}{\partial \xi_1}\delta_{i2}\delta_{k2} - \dfrac{\partial x_1}{\partial \xi_2}\delta_{i2}\delta_{k1} \ ,$$

$$(J_{ik})^{-1} = \frac{\varepsilon_{ki}\varepsilon_{ip}}{|J|}\frac{\partial x_p}{\partial \xi_l} \ . \tag{A-18}$$

The determinant of the Jacobian $|J|$ is expressed as:

$$|J| = \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial x_1}{\partial \xi_2} = \varepsilon_{ij}\frac{\partial x_i}{\partial \xi_1}\frac{\partial x_j}{\partial \xi_2} \ . \tag{A-19}$$

The first task is to obtain an expression for $\partial x_p/\partial \xi_l$ that can be substituted into equation (A-18). Regardless of the flow variable considered or its order of interpolation, the element boundaries are assumed curved to enable closer approximation of actual meniscus curvature. A second-order curvature is assumed which means that x_p can be expressed in terms of a quadratic interpolation with respect to the naturalized coordinates:

$$x_p = \Phi^{II}_N x_{Np} = \Phi^{II}_1 x_{1p} + \ldots + \Phi^{II}_9 x_{9p} \ . \tag{A-20}$$

The terms in equation (A-20) can be expanded using the second-order interpolation formulas given in equation (A-8) to yield:

$$\Phi^{II}_1 x_{1p} = \frac{1}{4}(\xi_1^2\xi_2^2 - \xi_1^2\xi_2 - \xi_1\xi_2^2 + \xi_1\xi_2)x_{1p} \ , \tag{A-21}$$

$$\Phi^{II}_2 x_{2p} = \frac{1}{2}(-\xi_1^2\xi_2^2 + \xi_1^2\xi_2 + \xi_2^2 - \xi_2)x_{2p} \ ,$$

$$\Phi^{II}_3 x_{3p} = \frac{1}{4}(\xi_1^2\xi_2^2 - \xi_1^2\xi_2 + \xi_1\xi_2^2 - \xi_1\xi_2)x_{3p} \ ,$$

$$\Phi^{II}_4 x_{4p} = \frac{1}{2}(-\xi_1^2\xi_2^2 - \xi_1\xi_2^2 + \xi_1^2 + \xi_1)x_{4p} \ ,$$

$$\Phi^{II}_5 x_{5p} = \frac{1}{4}(\xi_1^2\xi_2^2 + \xi_1^2\xi_2 + \xi_1\xi_2^2 - \xi_1\xi_2)x_{5p} \ ,$$

$$\Phi^{II}_6 x_{6p} = \frac{1}{2}(-\xi_1^2\xi_2^2 - \xi_1^2\xi_2 + \xi_2^2 + \xi_2)x_{6p} \ ,$$

$$\Phi^{II}_7 x_{7p} = \frac{1}{4}(\xi_1^2\xi_2^2 + \xi_1^2\xi_2 - \xi_1\xi_2^2 - \xi_1\xi_2)x_{7p} \ ,$$

$$\Phi^{II}_8 x_{8p} = \frac{1}{2}(-\xi_1^2\xi_2^2 + \xi_1\xi_2^2 + \xi_1^2 - \xi_1)x_{8p} \ ,$$

$$\Phi^{II}_9 x_{9p} = (\xi_1^2\xi_2^2 - \xi_2^2 - \xi_1^2 + 1)x_{9p} \ .$$

A generalized expression for x_p is obtained by rearranging the terms according to like products:

$$x_p = \frac{1}{4}\begin{pmatrix} a_p + b_p\xi_1 + c_p\xi_2 + d_p\xi_1\xi_2 + e_p\xi_1^2 \\ + f_p\xi_2^2 + g_p\xi_1\xi_2^2 + h_p\xi_1^2\xi_2 + p_p\xi_1^2\xi_2^2 \end{pmatrix} , \qquad (A\text{-}22)$$

where

$$a_p = 4x_{9p} ,$$

$$b_p = 2x_{4p} - 2x_{8p} ,$$

$$c_p = -2x_{2p} + 2x_{6p} ,$$

$$d_p = x_{1p} - x_{3p} + x_{5p} - x_{7p} ,$$

$$e_p = 2x_{4p} + 2x_{8p} - 4x_{9p} ,$$

$$f_p = 2x_{2p} + 2x_{6p} - 4x_{9p} ,$$

$$g_p = -x_{1p} + x_{3p} - 2x_{4p} + x_{5p} - x_{7p} + 2x_{8p} ,$$

$$h_p = -x_{1p} + 2x_{2p} - x_{3p} + x_{5p} - 2x_{6p} + x_{7p} ,$$

$$p_p = x_{1p} - 2x_{2p} + x_{3p} - 2x_{4p} + x_{5p} - 2x_{6p} + x_{7p} - 2x_{8p} + 4x_{9p} .$$

Differentiating the expression for x_p in equation (A-22) by ξ_l yields the following expression for the position derivative:

$$\frac{\partial x_p}{\partial \xi_l} = \frac{1}{4} A_{pl} , \qquad (A\text{-}23)$$

where

$$A_{p1} = b_p + d_p\xi_2 + 2e_p\xi_1 + g_p\xi_2^2 + 2h_p\xi_1\xi_2 + 2p_p\xi_1\xi_2^2 ,$$

$$A_{p2} = c_p + d_p\xi_1 + 2f_p\xi_2 + 2g_p\xi_2\xi_2 + h_p\xi_1^2 + 2p_p\xi_1^2\xi_2 .$$

Substitution of equation (A-23) into equations (A-18) and (A-19) yields complete expressions for the inverse and determinant of the Jacobian matrix:

$$(J_{ik})^{-1} = \frac{\varepsilon_{ip}\varepsilon_{kl}}{4|J|} A_{pl} , \qquad (A\text{-}24)$$

$$|J| = \frac{1}{16}(A_{11}A_{22} - A_{21}A_{12}) . \qquad (A\text{-}25)$$

2. Function Type 1

The next step is to calculate the derivative of the test function with respect to the naturalized coordinates. For the linear function, the general expression in equation (A-7) is expanded to yield:

$$\Phi_N^I = \frac{1}{4} (\xi_{N1} \xi_{N2} \xi_1 \xi_2 + \xi_{N1} \xi_1 + \xi_{N2} \xi_2 + 1) \ . \tag{A-26}$$

The corresponding partial derivative is:

$$\frac{\partial \Phi_N^I}{\partial \xi_k} = \frac{1}{4} (\xi_{Nk} + \xi_{N1} \xi_{N2} (\xi_1 \delta_{2k} + \xi_2 \delta_{1k})) \ . \tag{A-27}$$

The global derivative $\partial \Phi_N{}^I / \partial x_i$ is obtained by combining equations (A-24) and (A-27) to yield:

$$\frac{\partial \Phi_N^I}{\partial x_i} = \frac{\varepsilon_{ip}}{16|J|} [-(\xi_{N2} + \xi_1 \xi_{N1} \xi_{N2}) A_{p1} + (\xi_{N1} + \xi_2 \xi_{N1} \xi_{N2}) A_{p2}] \ . \tag{A-28}$$

3. Function Type 3

The gradient of the quadratic function is best solved by considering the corner, midside, and center nodes separately. For the corner nodes, the expression for $\Phi_N{}^{II}$ in equation (A-8a) is split into three terms:

$$\Phi_N^{II} = C_N E_1 E_2 E_3 \ , \tag{A-29}$$

where

$$C_N = \frac{1}{4} \xi_{N1} \xi_{N2} \ ,$$

$$E_1 = \xi_1 \xi_2 \ ,$$

$$E_2 = \xi_{N1} \xi_1 + 1 \ ,$$

$$E_3 = \xi_{N2} \xi_2 + 1 \ .$$

The general form of the derivative is then:

$$\Phi_{N,i}^{II} = C_N (E_{1,i} E_2 E_3 + E_1 E_{2,i} E_3 + E_1 E_2 E_{3,i}) \ , \tag{A-30}$$

where

$$E_{1,i} = (J_{ik})^{-1} \frac{\partial E_1}{\partial \xi_k} = (J_{ik})^{-1} (\xi_1 \delta_{k2} + \xi_2 \delta_{k1}) \ ,$$

$$E_{2,i} = (J_{ik})^{-1} \frac{\partial E_2}{\partial \xi_k} = (J_{ik})^{-1} \xi_{N1} \delta_{k1} \ ,$$

$$E_{3,i} = (J_{ik})^{-1} \frac{\partial E_3}{\partial \xi_k} = (J_{ik})^{-1} \xi_{N2} \delta_{k2} \ .$$

Further rearranging of the E-terms in equation (A-30) yields:

$$\Phi_{N,i}^{II} = C_N(J_{ik})^{-1}(C_1\delta_{1k}+C_2\delta_{2k}) \ , \qquad N = 1, 3, 5, \text{ and } 7 \quad \text{(A-31a)}$$

where

$$C_N = \frac{1}{4}\, \xi_{N1}\xi_{N2} \ ,$$

$$C_1 = \xi_2(\xi_{N2}\xi_2+1)(2\xi_{N1}\xi_1+1) \ ,$$

$$C_2 = \xi_1(\xi_{N1}\xi_1+1)(2\xi_{N2}\xi_2+1) \ .$$

For the midside nodes, equation (A-8b) is expressed in the following general form:

$$\Phi_N^{II} = \frac{\xi_{N\alpha}}{2}\,(1-\xi_\beta^2)(\xi_{N\alpha}\xi_\alpha^2+\xi_\alpha) \ . \tag{A-32}$$

For $N = 2$ or 6, $\alpha = 2$ and $\beta = 1$. If $N = 4$ or 8, $\alpha = 1$ and $\beta = 2$. The local derivative of equation (A-32) becomes:

$$\frac{\partial\Phi_N^{II}}{\partial\xi_k} = \frac{\xi_{N\alpha}}{2}\,[-2\xi_\beta(\xi_{N\alpha}\xi_\alpha^2+\xi_\alpha)\delta_{\beta k}+(1-\xi_\beta^2)(2\xi_{N\alpha}\xi_\alpha+1)\delta_{\alpha k}] \ . \tag{A-33}$$

When substituted into equation (A-17), the following general expression is obtained.

$$\Phi_{N,i}^{II} = C_N(J_{ik})^{-1}[C_\beta\delta_{\beta k}+C_\alpha\delta_{\alpha k}) \ , \qquad \begin{array}{l} N = 2 \text{ and } 6; \ \alpha = 2 \text{ and } \beta = 1 \\ N = 4 \text{ and } 8; \ \alpha = 1 \text{ and } \beta = 2 \end{array} \quad \text{(A-31b)}$$

where

$$C_N = \frac{\xi_{N\alpha}}{2} \ ,$$

$$C_\beta = -2\xi_\beta(\xi_{N\alpha}\xi_\alpha^2+\xi_\alpha) \ ,$$

$$C_\alpha = (1-\xi_\beta^2)(2\xi_{N\alpha}\xi_\alpha+1) \ .$$

For the center node, one differentiates equation (A-8c) by ξ_k to obtain the local derivative:

$$\frac{\partial\Phi_N^{II}}{\partial\xi_k} = -2[\xi_1(1-\xi_2^2)\delta_{1k}+\xi_2(1-\xi_1^2)\delta_{2k}] \ . \tag{A-34}$$

Substituting equation (A-34) into equation (A-19) yields:

$$\Phi_{N,i}^{II} = -2(J_{ik})^{-1}[C_1\delta_{1k}+C_2\delta_{2k}] \ , \qquad N = 9 \qquad \text{(A-31c)}$$

where

$$C_1 = \xi_1(1-\xi_2^2) \ ,$$

$$C_2 = \xi_2(1-\xi_1^2) \ .$$

4. Function Type 5

Derivation of the gradient of the fourth-order interpolation function requires lengthy algebraic manipulation. To summarize, one expresses it in general terms:

$$\Phi_{N,i}^{IV} = C_N(J_{ik})^{-1}(C_1\delta_{1k}+C_2\delta_{2k}) \ . \tag{A-35a}$$

For the off-axes nodes, one has:

$$C_N = 1/36 \ , \qquad N = 1, 5, 9, \text{ and } 13$$

$$C_N = -8/18 \ , \qquad N = 2, 4, 6, 8, 10, 12, 14, \text{ and } 16$$

$$C_N = 1/36 \ , \qquad N = 17, 19, 21, \text{ and } 23$$

$$C_1 = (16\xi_{N1}^2\xi_1^3+12\xi_{N1}^3\xi_1^2-2\xi_1-\xi_{N1})(4\xi_{N2}^2\xi_2^4+4\xi_{N2}^3\xi_2^3-\xi_2^2-\xi_{N2}\xi_2) \ ,$$

$$C_2 = (16\xi_{N2}^2\xi_2^3+12\xi_{N2}^3\xi_2^2-2\xi_2-\xi_{N2})(4\xi_{N1}^2\xi_1^4+4\xi_{N1}^3\xi_1^3-\xi_1^2-\xi_{N1}\xi_1) \ .$$

For the center node ($N = 25$), one has:

$$C_N = 16 \ , \qquad N = 25$$

$$C_1 = \left(4\xi_1^3-\frac{5}{2}\xi_1\right)\left(\xi_2^4-\frac{5}{4}\xi_2^2+\frac{1}{4}\right) \ ,$$

$$C_2 = \left(4\xi_2^3-\frac{5}{2}\xi_2\right)\left(\xi_1^4-\frac{5}{4}\xi_1^2+\frac{1}{4}\right) \ .$$

For the remaining nodes, one uses the general expression:

$$\Phi_{N,i}^{IV} = C_N(J_{ik})^{-1}(C_\alpha\delta_{\alpha k}+C_\beta\delta_{\beta k}) \ , \tag{A-35b}$$

and the coefficients

$$C_N = 2/3 \ , \qquad \alpha = 1 \ / \ \beta = 2 \ , \qquad N = 3 \text{ and } 11$$

$$C_N = -32/3 \ , \qquad \alpha = 1 \ / \ \beta = 2 \ , \qquad N = 18 \text{ and } 22$$

$$C_N = 2/3 \ , \qquad \alpha = 2 \ / \ \beta = 1 \ , \qquad N = 7 \text{ and } 15$$

$$C_N = -32/3 \ , \qquad \alpha = 2 \ / \ \beta = 1 \ , \qquad N = 20 \text{ and } 24$$

$$C_\alpha = \left(4\xi_\alpha^3-\frac{5}{2}\xi_\alpha\right)\left(4\xi_{N\beta}^2\xi_\beta^4+4\xi_{N\beta}^3\xi_\beta^3-\xi_\beta^2-\xi_{N\beta}\xi_\beta\right) \ ,$$

$$C_\beta = \left(\xi_\alpha^4-\frac{5}{4}\xi_\alpha^2+\frac{1}{4}\right)\left(16\xi_{N\beta}^2\xi_\beta^3+12\xi_{N\beta}^3\xi_\beta^2-2\xi_\beta^2-\xi_{N\beta}\right) \ .$$

5. Function Type 2

For the type 2 functions, one still employs the general relationship in equations (A-31a) and (A-35a). For the meniscus elements, type 2a, one has:

$$C_N = 1/2 \ , \qquad\qquad N = 3 \text{ and } 6$$

$$C_1 = \xi_{N1}(1-\xi_2^2) \ ,$$

$$C_2 = -2\xi_2(\xi_{N1}\xi_1+1) \ ,$$

$$C_N = 1/4 \ , \qquad\qquad N = 1, 2, 4, \text{ and } 5$$

$$C_1 = \xi_{N1}(\xi_2^2+\xi_{N2}\xi_2) \ ,$$

$$C_2 = (2\xi_2+\xi_{N2})(\xi_{N1}\xi_1+1) \ .$$

For the sidewall elements, type 2b, one has:

$$C_N = 1/2 \ , \qquad\qquad N = 2 \text{ and } 5$$

$$C_1 = -2\xi_1(\xi_{N2}\xi_2+1) \ ,$$

$$C_2 = \xi_{N2}(1-\xi_1^2) \ ,$$

$$C_N = 1/4 \ , \qquad\qquad N = 1, 3, 4, \text{ and } 6$$

$$C_1 = (2\xi_1+\xi_{N1})(\xi_{N2}\xi_2+1) \ ,$$

$$C_2 = \xi_{N2}(\xi_1^2+\xi_{N1}\xi_1) \ .$$

6. Function Type 4

Again, one uses the general expression for the derivative in equations (A-31a) and (A-35a), and calculates the coefficients for the combined second-fourth order functions. For the meniscus elements and the off-axes nodes, one has:

$$C_N = 1/2 \ , \qquad\qquad N = 1, 3, 7, \text{ and } 9$$

$$C_N = -8/6 \ , \qquad\qquad N = 4, 6, 10, \text{ and } 12$$

$$C_1 = (4\xi_{N2}^2\xi_2^4+4\xi_{N2}^3\xi_2^3-\xi_2^2-\xi_{N2}\xi_2)(2\xi_1+\xi_{N1}) \ ,$$

$$C_2 = (16\xi_{N2}^2\xi_2^3+12\xi_{N2}^3\xi_2^2-2\xi_2-\xi_{N2})(\xi_1^2+\xi_{N1}\xi_1) \ .$$

For the noncenter nodes lying on the ξ_2-axis:

$$C_N = 1/6 \ , \qquad\qquad N = 2 \text{ and } 8$$

$$C_N = -8/3 \ , \qquad\qquad N = 13 \text{ and } 15$$

$$C_1 = -2\xi_1(4\xi_{N2}^2\xi_2^4 + 4\xi_{N2}^3\xi_2^3 - \xi_2^2 - \xi_{N2}\xi_2) \ ,$$

$$C_2 = (16\xi_{N2}^2\xi_2^3 + 12\xi_{N2}^3\xi_2^2 - 2\xi_2 - \xi_{N2})(1 - \xi_1^2) \ .$$

For the noncenter nodes lying on the ξ_1-axis:

$$C_N = 2 \ , \qquad\qquad N = 5 \text{ and } 11$$

$$C_1 = \left(\xi_2^4 - \frac{5}{4}\xi_2^2 + \frac{1}{4}\right)(2\xi_1 + \xi_{N1}) \ ,$$

$$C_2 = \left(4\xi_2^3 - \frac{5}{2}\xi_2\right)(\xi_1^2 + \xi_{N1}\xi_1) \ .$$

For the center node, one has:

$$C_N = 4 \ , \qquad\qquad N = 14$$

$$C_1 = -2\xi_1\left(\xi_2^4 - \frac{5}{4}\xi_2^2 + \frac{1}{4}\right) \ ,$$

$$C_2 = \left(4\xi_2^3 - \frac{5}{4}\xi_2\right)(1 - \xi_1^2) \ .$$

The expressions for the sidewall elements are very similar but are based on a different relationship between the local nodes and naturalized coordinate frame. For the off-axes nodes, one has:

$$C_N = 1/12 \ , \qquad\qquad N = 1, 5, 7, \text{ and } 11$$

$$C_N = -8/6 \ , \qquad\qquad N = 2, 4, 8, \text{ and } 10$$

$$C_1 = (16\xi_{N1}^2\xi_1^3 + 12\xi_{N1}^3\xi_1^2 - 2\xi_1 - \xi_{N1})(\xi_2^2 + \xi_{N2}\xi_2) \ ,$$

$$C_2 = (4\xi_{N1}^2\xi_1^4 + 4\xi_{N1}^3\xi_1^3 - \xi_1^2 - \xi_{N1}\xi_1)(2\xi_2 + \xi_{N2}) \ .$$

For the noncenter nodes lying on the ξ_1-axis:

$$C_N = 1/6 \ , \qquad\qquad N = 6 \text{ and } 12$$

$$C_N = -8/3 \ , \qquad\qquad N = 13 \text{ and } 15$$

$$C_1 = (16\xi_{N1}^2\xi_1^3 + 12\xi_{N1}^3\xi_1^2 - 2\xi^2 - \xi_{N1})(1 - \xi_2^2) \ ,$$

134

$$C_1 = -2\xi_2(4\xi_{N1}^2\xi_1^4 + 4\xi_{N1}^3\xi_1^3 - \xi_1^2 - \xi_{N1}\xi_1) \ .$$

For the noncenter nodes lying on the ξ_2-axis:

$$C_N = 2 \ , \qquad\qquad N = 3 \text{ and } 9$$

$$C_1 = \left(4\xi_1^3 - \frac{5}{2}\xi_1\right)(\xi_2^2 + \xi_{N2}\xi_2) \ ,$$

$$C_2 = \left(\xi_1^4 - \frac{5}{4}\xi_1^2 + \frac{1}{4}\right)(2\xi_2 + \xi_{N2}) \ .$$

For the center node, one again has:

$$C_N = 4 \ , \qquad\qquad N = 14$$

$$C_1 = \left(4\xi_1^3 - \frac{5}{2}\xi_1\right)(1 - \xi_2^2) \ ,$$

$$C_2 = -2\xi_2\left(\xi_1^4 - \frac{5}{4}\xi_1^2 + \frac{1}{4}\right) \ .$$

E. Second-Order Gradients

1. General Form

Second-order gradients arise in the velocity terms of the SIMPLER pressure correction equation. Both dyadic and Laplacian forms (i.e., $\Phi_N{}^{II}{}_{,ij}$ and $\Phi_N{}^{II}{}_{,ii}$, respectively) must be considered for accurate pressure estimates. Since the gradients pertain to velocity, only the second-order gradient of the type 3 function need be considered. To derive an expression for the most general form, $\Phi_N{}^{II}{}_{,ij}$, one begins by treating $\Phi_N{}^{II}{}_{,i}$ as the function to be differentiated in equation (A-17). That is:

$$\Phi_{N,ij}^{II} = (J_{js})^{-1}\frac{\partial}{\partial\xi_s}(\Phi_{N,i}^{II}) \ . \tag{A-36}$$

Substitution of the expression for $\Phi_N{}^{II}{}_{,i}$ from equation (A-17) into equation (A-36) yields:

$$\Phi_{N,ij}^{II} = (J_{js})^{-1}\frac{\partial}{\partial\xi_s}\left((J_{ik})^{-1}\frac{\partial\Phi_N^{II}}{\partial\xi_k}\right) \ ,$$

which upon differentiation becomes:

$$\Phi_{N,ij}^{II} = (J_{js})^{-1}\left(\frac{\partial(J_{ik})^{-1}}{\partial\xi_s}\frac{\partial\Phi_N^{II}}{\partial\xi_k} + (J_{ik})^{-1}\frac{\partial^2\Phi_N^{II}}{\partial\xi_s\partial\xi_k}\right) \ . \tag{A-37}$$

The local derivative in compact tensor form can be represented by generalizing equation (A-33) as:

$$\frac{\partial \Phi_N^{II}}{\partial \xi_k} = C_N C_t \delta_{tk} \ .$$

(A-38)

Further application of the $\partial/\partial \xi_s$ operator yields:

$$\frac{\partial^2 \Phi_N^{II}}{\partial \xi_s \partial \xi_k} = C_N C_{ts}' \delta_{tk} \ ,$$

(A-39)

where

$$C_{ts}' = \frac{\partial C_t}{\partial \xi_s} \ .$$

The local derivative of the Jacobian inverse is expressed in a similar way. Differentiating the expression for $(J_{ik})^{-1}$ in equation (A-18) with respect to ξ_s, one obtains:

$$\frac{\partial (J_{ik})^{-1}}{\partial \xi_s} = \frac{\varepsilon_{ip} \varepsilon_{km}}{4|J|} A_{pms}' \ ,$$

(A-40)

where

$$A_{pms}' = \frac{\partial A_{pm}}{\partial \xi_s} \ ,$$

and

$$\frac{\partial |J|}{\partial \xi_s} = 0 \ .$$

Substitution of equations (A-39) and (A-40) into equation (A-37) yields:

$$\Phi_{N,ij} = (J_{js})^{-1} \frac{C_n \varepsilon_{ip} \varepsilon_{km}}{4|J|} (A_{pms}' C_k + A_{pms} C_{ks}') \ .$$

(A-41)

Upon expanding ε_{km} and the terms within parentheses, one obtains:

$$\Phi_{N,ij} = \frac{C_N \varepsilon_{ip}}{4|J|} \left[\begin{array}{l} (C_1 A_{p21}' - C_2 A_{p11}' + C_{11}' A_{p2} - C_{21}' A_{p1})(J_{j1})^{-1} \\ + (C_1 A_{p22}' - C_2 A_{p12}' + C_{12}' A_{p2} - C_{22}' A_{p1})(J_{j2})^{-1} \end{array} \right] \ .$$

(A-42)

Applying the $\partial/\partial \xi_s$ operator to A_{pm} in equation (A-23) yields an expression for A_{pms}'. The components of this tensor are:

$$A_{p11}' = 2e_p + 2h_p \xi_2 + 2p_p \xi_2^2 \ ,$$

(A-43)

$$A_{p12}' = A_{p21}' = d_p + 2g_p \xi_2 + 2h_p \xi_1 + 4p_p \xi_1 \xi_2 \ ,$$

$$A_{p22}' = 2f_p + 2g_p \xi_1 + 2p_p \xi_1^2 \ .$$

136

The C_k and C'_{ks} coefficients are obtained in a similar manner using the type 3 interpolation formulas defined in equation (A-31). For the corner nodes, one has:

$$C_1 = \xi_2(\xi_{N2}\xi_2+1)(2\xi_{N1}\xi_1+1) , \qquad N = 1, 3, 5, \text{ and } 7 \qquad \text{(A-44a)}$$

$$C_2 = \xi_1(\xi_{N1}\xi_1+1)(2\xi_{N2}\xi_2+1) ,$$

$$C'_{11} = 2\xi_{N1}\xi_2(\xi_{N2}\xi_2+1) ,$$

$$C'_{12} = C'_{21} = (2\xi_{N2}\xi_2+1)(2\xi_{N1}\xi_1+1) ,$$

$$C'_{22} = 2\xi_{N2}\xi_1(\xi_{N1}\xi_1+1) .$$

Formulation of the midside nodes follows in a similar manner. Note that for $N = 2$ or 6, $\alpha = 2$ and $\beta = 1$, while for $N = 4$ or 8, $\alpha = 1$ and $\beta = 2$.

$$C_\beta = -2\xi_\beta(\xi_{N\alpha}\xi_\alpha^2+\xi_\alpha) , \qquad N = 2 \text{ and } 6; \alpha = 2 \text{ and } \beta = 1 \qquad \text{(A-44b)}$$

$$C_\alpha = (1-\xi_\beta^2)(2\xi_{N\alpha}\xi_\alpha+1) , \qquad N = 4 \text{ and } 8; \alpha = 1 \text{ and } \beta = 2$$

$$C'_{\beta\beta} = -2\xi_\alpha(\xi_{N\alpha}\xi_\alpha+1) ,$$

$$C'_{\beta\alpha} = C'_{\alpha\beta} = -2\xi_\beta(2\xi_{N\alpha}\xi_\alpha+1) ,$$

$$C'_{\alpha\alpha} = 2\xi_{N\alpha}(1-\xi_\beta^2) .$$

For the center node:

$$C_1 = \xi_1(1-\xi_2^2) , \qquad\qquad N = 9 \qquad\qquad \text{(A-44c)}$$

$$C_2 = \xi_2(1-\xi_1^2) ,$$

$$C'_{11} = (1-\xi_2^2) ,$$

$$C'_{12} = C'_{21} = -2\xi_1\xi_2 ,$$

$$C'_{22} = (1-\xi_1^2) .$$

APPENDIX B

DESCRIPTION OF CIC COMPUTER PROGRAM

A. Introduction

The steady-state solutions presented in sections VI to VIII were obtained using the Combined Interfacial Convection (CIC) program, which was written and specifically tailored to meet the analytical needs of this study. The program employs the finite element equations derived in section IV and calculates the tensor coefficients for these equations from the basis functions derived in appendix A. The program includes a variety of data input and output options and permits considerable flexibility regarding computer system usage.

The main product of each CIC run is a unit 9 output file that lists the x_1 and x_2 coordinates, x_1 and x_2-component velocities, temperature, and dynamic pressure of each global node in the pore domain. These data are calculated in a five-loop iteration procedure from the parameters and logic options defined in a unit 7 input file.

CIC is written in Fortran 77 and was originally designed for interactive use on a Digital Equipment Corporation (DEC) VAX. However, it soon became apparent that the element density and interpolation orders required for accurate modeling of velocity, pressure, and temperature variation in the interline and sidewall regions necessitated use of a Cray-XMP. Although the program can be executed in an interactive mode, batch processing was used almost exclusively to generate the results in this study.

The Cray-XMP was accessed through NASA Marshall Space Flight Center's Engineering Analysis and Data System (EADS), which permitted simultaneous execution in several batch queues. The entire compiled version of the CIC code ordinarily took from 4 to 7 Megawords of memory to run depending on the initial dimension limits set in the program's internal arrays. The smallest cases consisted of 25 by 20 element domains (MXE = 25 and MYE = 20) while the largest were dimensioned for 34 by 25 elements (MXE = 34 and MYE = 25).

The following sections briefly describe the CIC program and provide general information useful for future users of the code. A complete copy of the CIC source code, including all unique subprograms needed to run with a standard Fortran 77 compiler, is given in appendix C.

B. Program Description

Apart from the convergence test for steady-state and the solution of meniscus geometry, each iteration loop of the CIC program is structured around the calculation of velocity, pressure or temperature using equations (88), (89), (83), (84), and (85). However, as was shown in appendix A, the coefficient tensors in these equations (equation (86)) are formulated and calculated in terms interpolation functions defined on the element-level. For example, the $M^1{}_{\alpha\beta}$ tensor used in the finite element equation for velocity is expressed in terms of the variable EA(e,n,m), where e = global element number, n = local element corresponding to α, and m = local element corresponding to β. Since several local nodes can share the same global node location, construction

of the coefficients in each finite element equation requires summing the contributions from all elements in contact with the global node.

For velocity, velocity correction, and temperature (equations (88), (89), and (84), respectively), the equation for each global node is solved sequentially until a convergence test involving all node values is satisfied. The appropriate contributions from the locally defined nodes and tensor coefficients are obtained via application of an association matrix $CE(n_g, e_g)$. n_g corresponds to the global node number, while e_g identifies the global elements contacting and contributing to the finite element equation at n_g. With the quadrilateral element geometry used in this study, the maximum number of elements that can contribute to a single global node is four. The number of elements e_g associated with each n_g value is stored in $NCE(n_g)$.

With the pressure correction (equations (83) and (85)), however, all the finite element coefficients are constructed concurrently, and node pressures are solved using Gauss-Jordan elimination. The coefficients associated with each global node are obtained by sequentially adding local node contributions on an element-by-element basis.

In section IV and appendix A, the node configurations used to model parameter variation in the element domain were defined. Each of these local nodes, in turn, has a global counterpart which is referenced by the coefficients in the finite element equations for velocity, pressure and temperature. Regardless of the method for summing the local contributions, a consistent matching between the local and global numbering schemes is required.

For velocity, type 3 elements (fig. 6) are employed throughout the domain and nine local nodes per element. The approach for numbering the global nodes is best exemplified by the 3 by 3 element domain on the left-hand side of figure B-1. All elements are defined by their global reference e. The global node n_g associated with any element's local node n_l is given by $n_g = CQ(e, n_l)$. $CQ(e, n_l)$ and $CE(n_g, e_g)$ are used to construct finite element coefficients for velocity.

Nominal Type 3
Definition

Additional for Type
4 and 5 Elements
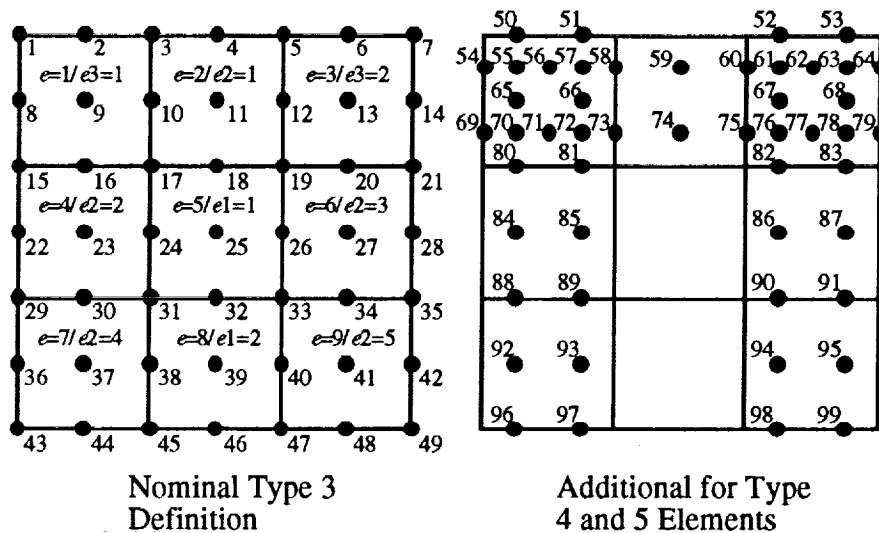
Figure B-1. Global node numbering scheme (velocity and temperature).

140

The solution domain for temperature is similar to that for velocity. Here, however, one employs type 5 elements in the upper corners, type 4a elements for noncorner elements along the meniscus, type 4b elements in noncorner elements along the sidewall and type 3 elements throughout the rest of the domain. Referring again to figure B-1, the domain consists of 99 global nodes—the first 49 of which coincide with the definition used for velocity. The additional 50 nodes in the right-hand diagram account for the extra local nodes needed for the type 4 and 5 elements.

Because of the variety of element types, a more complex approach to the global/local node association must be employed. The coincidence matrix that matches the local node number to its appropriate global value is embodied in the program variables $CQ(e,n_1)$, $C24(e_2,n_2)$ and $C44(e_3,n_3)$. When summing type 3 element contributions, which is done throughout the interior elements, the global node is specified directly by the value of $CQ(e,n_1)$, where $e \equiv$ global element number and $n_1 \equiv$ local node number. When the summing involves type 4 or 5 elements, the C24 and C44 arrays are used. Unlike CQ, the element number index for these arrays matches the ordering of type 4 and 5 elements in the domain, namely e_2 and e_3. Translation of the e designation, which is used to identify element contributions with $CE(n_g,e_g)$, into e_2 and e_3 is accomplished through the array $ECL(e,m)$. $ECL(e,1)$ denotes the element type, where $ECL(e,1) = 0$, 1 or 2 specifies whether e is a type 3, 4 or 5 element, respectively. For e values in which $ECL(e,1) \neq 0$, $ECL(e,2)$ represents the e_2 or e_3 value corresponding to e.

The coincidence matrices are used primarily to relate the locally defined tensor coefficients to the global nodes referenced in the finite element equations. An example is the designation of the global node corresponding to the first sidewall element's ($e = 4$ and $e_2 = 2$) fifth local node ($n_2 = 5$). The global node is obtained from the compound expression $C24(ECL(4,2),5) = 31$, where $ECL(4,1) = 1$ (type 4 element) and $ECL(4,2) = 2$ (e_2 designation). Similarly, the global node for the fourth local node ($n_3 = 4$) of the second corner element ($e = 3$ and $e_3 = 2$) is determined from $C44(ECL(3,2),4) = 83$, where $ECL(3,1) = 2$ (type 5 element) and $ECL(3,2) = 2$ (e_3 designation). Note that element types 4a and 4b are distinguished by whether $e \leq$ NXE (number of elements in x_1-direction) or $e >$ NXE, respectively.

The same type of bookkeeping procedure is used with pressure. Here, however, the solution routine is based on the simpler global geometry shown in figure B-2.
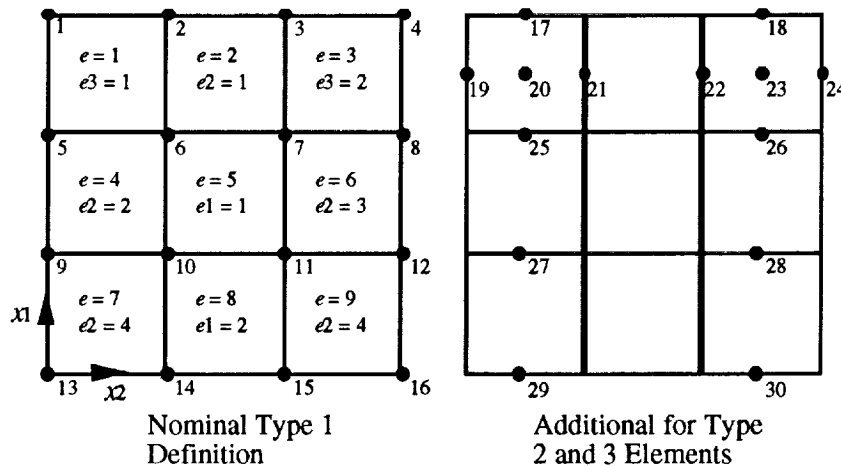
Figure B-2. Global node numbering scheme (pressure).

The ECL(e,1) values of 0, 1 and 2 now correspond to type 1, 2 and 3 elements, and the coincidence arrays associated with these elements are CL(e_1,n_1), CLLQ(e_2,n_2) and CLQ(e_3,n_3), respectively. CLLQ and CLQ play the same role as C24 and C44 before. However, CL is referenced according to the label e_1 which represents the ordering of type 1 elements.

Although the Gauss-Jordan solution routine employs the node definition shown in figure B-2, CIC output of flow parameter values (including pressure), and the CL, CLLQ, and CLQ coincidence matrices are based on the velocity-node definition in figure B-1. To construct the finite element coefficients for pressure, CIC employs an additional array, CNODE(n_v), which associates the velocity-based global node n_v to that of pressure n_p by n_p = CNODE(n_v). Note that most of the n_v-values lack an associated n_p designation. At the completion of each pressure iteration, the results are recast in terms of the velocity reference via n_v = GNODE(n_p).

The manipulation and use of coincidence arrays is potentially the most confusing aspect of the CIC program. The rest of the algorithms and procedures are more straightforward and are documented in the comment blocks of the source code listed in appendix C. To facilitate future reference, a summary of key program variables, excluding those included in the data read in statements, is provided in table B-1.

Table B-1. Key CIC program variables.

Code Parameter	I/D*	Description
MXE, MYE	I	Specifies maximum number of elements in x_1 and x_2 directions for code sizing purposes. Specified in PARAMETER statement.
XI(m,n)	D	ξ_1 and ξ_2 absissae values for Gaussian quadrature integration. m ≡ NPOINT + 1. n ≡ ξ_1 or ξ_2 direction.
W(m)	D	Ordinate values corresponding to XI for Gaussian quadrature integration. m ≡ NPOINT + 1.
A1(m) to D1(m)	D	ap to dp coefficients for 1st order approximation between xp = xp(xk).
XINL(m,n)	D	Naturalized coordinates for four local nodes in type 1 element. m ≡ local node number. n ≡ ξ_1 or ξ_2 value.
XINLQ(m,n,p)	D	Naturalized coordinates for six local nodes in type 2 elements. m ≡ local node number. n ≡ ξ_1 or ξ_2 value. p ≡ type 2a or 2b element.
DETJL	D	Jacobian determinant for type 1 element.
A2(m) to P2(m)	D	ap to pp coefficients for 2nd order approximation between xp = xp(xk).
JINV(i,k)	D	Inverse Jacobian tensor for type 3 element. (Jik)-1.
XG(m,n)	D	x_1 and x_2 values for each global node. m ≡ number of global node. n ≡ x_1 or x_2 value.
XINQ	D	Naturalized coordinates for nine local nodes in type 3 element.
DETJQ	D	Jacobian determinant for type 3 element.

* I = User Controlled (Independent)/Determined by CIC (Dependent)

Code Parameter	I/D	Description
VEL(m,n)	D	x_1 and x_2 velocity components for each global node. m ≡ number of global node. n ≡ x_1 or x_2 value.
PSI(m)	D	Stream function values at each global node. m ≡ number of global node.
SURFL	D	Surface length for each boundary element. Four boundaries.
SURFN	D	x1 and x2 normal components corresponding for SURFL element surfaces.
SURFT	D	x1 and x2 tangent components corresponding to SURFL element surfaces.
CL(m,n)	D	Global node number (global-local node coincidence matrix) for type 1 element definition. m ≡ element number. n = local node number.
CLLQ(m,n)	D	Global node number (global-local node coincidence matrix) for type 2 element definition. m ≡ element number. n = local node number.
CLQ(m,n)	D	Global node number (global-local node coincidence matrix) for type 3 element definition. m ≡ element number. n = local node number.
CQ(m,n)	D	Global node number (global-local node coincidence matrix) for type 3 element definition. m ≡ element number. n = local node number.
ECL(m,n)	D	Association matrix
XSURF(m)	D	x1 positions for surface parameter calculations. m = NXP.
YSURF(m)	D	y(s) values corresponding to XSURF values. m = NXP.
YSURF0(m)	D	y(s) values for pure isothermal, static meniscus solution.
PSURF(m)	D	Pd values corresponding to XSURF values.
PBUB(m)	D	Corrected bubble points. = Pvapor - PSURF
TSURF(m)	D	Surface temperature values corresponding to XSURF.
SURFNN(m,n)	D	Surface normal vectors at points corresponding to XSURF. m = NXP. n = n1 or n2.
SURFNT(m,n)	D	Surface tangent vectors at points corresponding to XSURF. m = NXP. n = n1 or n2.
IMODE	I	Computer system operation mode. = 0: VAX batch mode, = 1: VAX interactive mode, and = 2: CRAY batch mode.

Code Parameter	I/D	Description
NLNQ	I	Number of local nodes for type 3 element (9).
NLNL	I	Number of local nodes for type 1 element (4).
TPARAM	I	Time step factor h. h = 0.5 for Crank-Nicholson scheme.
ACCEL(m)	I	Acceleration with respect to Earth gravity. m = components in -x1 and -x2 direction.
QIN(m)	I	Heat input rate (disabled).
SB(i,k)	I	2nd-order surface variable integral matrix. $= 30(\Omega_{ik}^{*3})/l^e$ in Eq. 4.42.
SB2(i,k)	I	4th-order surface variable integral matrix. $= 5670(\Omega_{ik}^{*3})/l^e$ in Eq. 4.42.
SC(i)	I	2nd-order surface variable integral matrix. $= 6(\Omega_{i}^{*2})/l^e$ in Eq. 4.35.
SC2(i)	I	4th-order surface variable integral matrix. $= 90(\Omega_{i}^{*2})/l^e$ in Eq. 4.35.
NXP, NYP	D	Number of global nodes in x1 and x2 direction. NXP = 2NXE + 1.
NGNODE	D	Number of global nodes based on type 3 element definition for all elements.
NELEMENT	D	Total number of elements.
NROW, NCOL	D	2 x number of global nodes along top and side of domain.
NGNODE2	D	Total number of global nodes. Includes 4th-order temperature elements.
RA	D	Rayleigh number. = BO*MAIN*RVOLSURF.
VREC	D	Recoil parameter. = CA*(FRHO-1)/REVAPIN**2.
NLNS	D	Number of local nodes for sidewall elements in pressure calculation.
NLNC	D	Number of local nodes for corner elements in pressure calculation.

144

B. Input/Output Structure

CIC was originally developed to run interactively on the VAX, or in batch mode on either the VAX or CRAY. These three options are distinguished by the integer flag IMODE which is controlled via a DATA statement in the CAV module. IMODE = 0, 1, and 2 specifies an input/output format compatible with VAX batch, VAX interactive and CRAY batch operations, respectively. Nearly all of the steady-state parameter solutions obtained in this study were performed in the CRAY batch mode, where IMODE = 2. This approach was preferred due to the sizable memory requirements of the program.

The input/output file structure for this mode of operation is illustrated in figure B-3. This figure identifies all the files needed to run a case. It consists of three input and four output files.

Input	CIC	Output
Unit 1 Command for Unit 9-typ graphics data dump into Unit 1 at end of next steady-sta iteration • "large3.ts2"		**Unit 3** Specified parameter set after each time step • "large3.dat"
Unit 2 Command for program termination and output dump into Unit 9 file • "large3.tst"	**CAV**	**Unit 8** General output of specified parameter values • "output.dat"
Unit 7 Main input data file • "input.dat"		**Unit 9** Graphics output. Steady-state parameter values at all nodes • "plot.dat"
		Unit 10 Graphics output at end of time step. Enabled by Unit 1 file. • "clrg3.dat"

Figure B-3. CIC input/output file structure.

The unit 7 input file is required to execute the CIC program and contains critical data, such as parameter values, input/output control flags and heating regime. An example of this file is shown in table B-2 along with a variable description list in table B-3. When IMODE = 2, the unit 7 file is designated as "input.dat" prior to all read in or data manipulation statements.

Table B-2. CIC unit 7 input file.

```
          1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123456789012
Case 1C01
Pr    lambda    frho    Bo    Bie    Ma    Rs    Cr
e0    e-1       e+3     e-6   e+1    e+3   e-1   e-8


HUGE3
DATA SET:  FILE SPECIFICATION
Unit 8 output file flag (IOUT8)                      1
Unit 9 output file flag (IOUT9)                      1
Sidewall heating option (ITYPE)                      0
Press interp func option (IPRESS)                    1
Surf param inclusion option (ICURVE)                 7
Meniscus symmetry flag (ISYMM)                       1
Steady-state init temp flag (ISSTEMP)               1


DATA SET:  PROPERTY DATA
Contact angle (CANG)                        1.5000E+01
Bond number (BO)                            1.0000E-06
Capillary number-disabled (CA)              1.0000E-20
Marangoni number (MA)                       1.0000E+03
Vol to surf resp ratio (RVOLSURF)           1.0000E-01
Prandtl number (PR)                         1.0000E+00
Evaporation resistance (REVAP)              1.0000E+03
Crispation number (CR)                      1.0000E-08
Liq dens/vapor dens (FRHO)                  1.0000E+03
Dimensionless lat heat (LHS)                1.0000E+30
Biot number (BI)                            1.0000E+02
Reference corner press (PCORNER)            8.0000E+05
Initial liq temperature (TINIT)             1.0000E+00
Aspect ratio- height/width (RASPECT)        1.0000E+00


DATA SET:  CONVERGENCE CRITERIA
Vel conv error tolerance (VTOL)             1.0000E-03
Press conv error tolerance (PTOL)           2.0000E-03
Temp conv error tolerance (TTOL)            5.0000E-05
Steady-state conv tolerance (SSTOL)         1.0000E-04
Meniscus conv tolerance (MENTOL)            1.0000E-03
Max time increment (DELTMAX)                1.0000E-04


DATA SET:  STEP AND INTERVAL DATA
Maximum number of time steps (NTSTEP)             8000
Number elements in X1-direction (NXE)               35
Number of side elements (NSMALL(1))                  0
Scaling ratio (FSCALE(1))                   2.0000E+00
Number elements in X2-direction (NYE)               30
```

```
Number of side elements (NSMALL(2))                          0
Scaling ratio (FSCALE(2))                            2.0000E+00
Number quadrature integ pts (NPOINT)                         3
Meniscus integ interval size (NINT)                      10000
# Marangoni number ramp-up steps                            1
# Revap resistance ramp-up steps                            1

DATA SET:  PLOT AND OUTPUT SPECIFICATIONS
Nodes printed out for          1    2    3    4    5    6
Interactive Display           72   73   74   75   76   77
                             143  144  145  146  147  148
                             214  215  216  217  218  219
                             285  286  287  288  289  290
                             356  357  358  359  360  361
Number of isotherms (unit 9 only)                    20
Number of streamlines                                20
Number of isobars                                    80
Plot velocity scale (unit 9 only)            3.0000E-01
Plot arrow scale                             2.0000E-02
Plot arrow angle                             2.0000E+01
```

Table B-3. CIC input file variable description.

Parameter	Type	Description
lines 1 and 2	-	Space location reference. Skipped by CAV read statement.
lines 3 to 5	C	User title and file description. Copied onto unit 8 output file.
line 6	-	Skipped by CAV read statement.
QUETYPE	C	Queue type specification for batch execution on the NASA MSFC EADS1 Cray-XMP. QUETYPE controls read in of unit 1 and 2 files, and creation unit 3 and 10 files. Possible queue types are "HUGE3", "HUGE2", "LARGE4", "LARGE3" AND "LARGE2".
IOUT8	I	Unit 8 output option flag. = 0: no unit 8 output, = 1: abbreviated unit 8 output, and = 2: full-size unit 8 output.
IOUT9	I	Unit 9 output option flag. = 0: no unit 9 output, and = 1: unit 9 output created at completion of meniscus convergence.
ITYPE	I	Heating mode option flag. = 0: superheated, isothermal conditions along boundaries 1, 2 and 3, and = 1: subcooled conditions over the same boundaries.
IPRESS	I	Option flag dictating pressure interpolation order in corner, sidewall and meniscus elements. = 0: 1st-order (type 1) elements throughout domain. = 1: type 3 in corners and type 2 along sidewall and meniscus.
ICURVE	I	Variable inclusion option for meniscus calculation. = 0: hydrostatic pressure (Bo term). = 1: Bo and Pd terms. = 2: Bo and recoil (Vr) terms. = 3: Bo and surface tension variation (Cr) terms. = 4: Bo, Pd and Vr terms. = 5: Bo, Pd and Cr terms. = 6: Bo, Vr and Cr terms. = 7: Bo, Pd, Vr and Cr terms. = 8: Viscous stress terms only. = 9: All surface variables.
ISYMM	I	Symmetry option. = 1: meniscus integration performed on only one side.
ISSTEMP	I	Initial basic state temperature flag. = 1: basic state used as initial temperature distribution.
CANG	R	Contact angle ω (deg)
BO	R	Bond number Bo
CA	R	Capillary number Ca. Calculated from $Ca = CR/Ma$ if $Ma \neq 0$. Input value of Ca used if $Ma = 0$.
MAIN	R	Marangoni number Ma.
RVOLSURF	R	Thermal response ratio λ.
PR	R	Prandtl number Pr
REVAPIN	R	Interfacial resistance Rs
CR	R	Crispation number Cr
FRHO	R	Ratio liquid/vapor density f_ρ.
LHS	R	Latent heat parameter Λ.
BI	R	Biot number Bi.
PCORNER	R	Reference pressure in corners for calculation*
TINIT	R	Initial liquid temperature. Needed only if ISSTEMP $\neq$ 1.

148

RASPECT	R	Aspect ratio = 1.
VTOL	R	Tolerance for velocity calculation iteration.
PTOL	R	Convergence tolerance for pressure calculation iteration.
TTOL	R	Convergence tolerance for temperature calculation iteration.
SSTOL	R	Convergence tolerance for steady-state convergence.
MENTOL	R	Convergence tolerance for meniscus convergence.
DELTMAX	R	Initial estimate and maximum time step value.
NTSTEP	R	Maximum number of time steps in each steady-state iteration.
NXE, NYE	R	Number of elements in x_1 and x_2 directions. Controls size of numerical domain.
NSMALL(m)	I	Number of elements to scale width-wise along sidewalls (m = 1) and along meniscus (m = 2). Set = 0 for normal sizing.
FSCALE(m)	R	Scale factor used with NSMALL. Disabled when NSMALL(m) = 0.
NPOINT	I	Number of points for Gaussian quadrature integration.
NINT	I	Number of subintervals for meniscus integration and calculation. Δs calculated by dividing pore half-cavity width by NINT.
NMA	I	Number of time steps for Ma to reach specified MAIN value. Relaxes and eases convergence for high Ma.
NREVAP	I	Number of time steps for Rs to reach specified REVAPIN value. Relaxes and eases convergence for low Rs.
PNODE(m)	I	Global nodes printed out in unit 3 and 8 outputs. Must specify 36 node numbers in form shown in sample unit 7 input file. m ≡ global node number. PNODE(m) ≡ global node number.
NCONTOUR	I	Dummy variable used to copy required graphics program data from unit 7 to unit 9. Represents number of isotherms, streamlines and isobars for plot package.
XARROW	R	Dummy variable used to copy required graphics program data from unit 7 to unit 9. Designates arrow properties for velocity vector graphics.

The other two input files enable review of iteration progress and a limited degree of program control during batch execution. The name of the file that is read in depends on the QUETYPE assignment in the unit 7 file. For example, the names given in figure B-3 pertain to a case run in the "large3" queue. Note that the name used for the queue depends entirely on the system used. These correspond to the designations used by the EADS1 system at NASA MSFC.

The unit 1 file consists of a single line containing a value of 0 or 1 for the variable ICHK, as shown in the example in table B-4. The file is checked at completion of every time step. Changing the value of ICHK to 1 causes CAV to print out a complete variable/node summary (unit 9-type output) which is automatically dumped into the unit 10 output file. Note that the unit 10 output is also generated at the end of every steady-state iteration. If ICHK = 1, ICHK is reset to 0 at the completion of the unit 10 data dump.

Table B-4. CIC unit 1 input file.

```
Graphics printout? (=0:no; =1:yes)                    0
```

The unit 2 file also consists of a single line but specifies the value of the variable IGO. An example of this file is shown in table B-5. Ordinarily, IGO is set to 1. When IGO = 0, the program is

instructed to terminate program execution and dump the current output variable contents into the unit 9 output file. As with ICHK, the value of IGO in unit 2 is reset to 1 when this operation occurs.

Table B-5. CIC unit 2 input file.

```
Continuation flag (=1:continue; =0:stop)                    1
```

The most important output file from the standpoint of this study is the unit 9 file "plot.dat". This file contains a summary of the input variables read in through unit 7 and a complete summary of the velocity, temperature, stream function, pressure and node locations for each global node. Note that only the global nodes associated with the type 3 velocity-based definition are printed out. This file repeatedly provided the basic data used for graphics/plot applications, first-order surface deformation and other parameter calculations. The generation of the unit 9 output is enabled by setting IOUT9 = 1 in the unit 7 input file.

The unit 8 file provides a summary of velocity, temperature and pressure at the global nodes specified in the PNODE array in unit 7. A total of 36 values can be printed out at each time. The size of this file depends on the value of IOUT8 specified in unit 7. When IOUT8 = 2, an extensive summary showing the parameter values at the end of each time step is recorded onto unit 8. With IOUT8 = 1, the summary is briefer and only prints the values at the end of each meniscus iteration. When IOUT8 = 0, no unit 8 file is generated. This file was extremely useful during the development and testing of new algorithms and procedures in the CIC code.

An updated version of the unit 3 output file is created at the completion of each time step, and it presents the variable values at the nodes specified in the PNODE array. It also shows the convergence trends for all the iteration loop variables, including the meniscus and steady-state temperature. This file is extremely useful for monitoring program status and execution.

As was mentioned before, the unit 10 output file is created in response to the value of ICHK specified in unit 1. It is also generated after completion of steady-state convergence and before initiation of the next meniscus iteration step.

C. Program/Subprogram Structure

CIC consists of a main program module (CAV) and 25 subprograms. The relationship between CAV and its supporting subprograms is shown in figure B-4, and a listing of the entire program source code is provided in appendix C. Following is a description of the various subprogram elements.

CAV performs almost all of the general calculations in CIC. Its functions include: (1) file and data initialization and input, (2) velocity, pressure, temperature and meniscus iteration loops and convergence tests, and (3) data output. Callouts for all of the five main ancillary subprogram functions (i.e., domain geometric initialization, calculation of finite element equation coefficient tensors, meniscus calculation, stream function calculation and matrix solution) reside and are sequenced within the CAV module.

The primary role of GEOM is to establish the global node/local node coincidence arrays discussed in section B.2 and to calculate the coordinates for each global and local node. In addition,

CAV

GEOM   TENSCAL   MENISCUS   GAUSS

QCAL   GQWEIGHT   FUNC2   STREAM

GEOM2   DPDXL   EPS   CURVE

DPDXLQ   PHIL   FUNC

DPDXQ   PHILQ   FSIN

DPDX24   PHIQ

DPDX4   PHI24

DPDXDYQ   PHI4
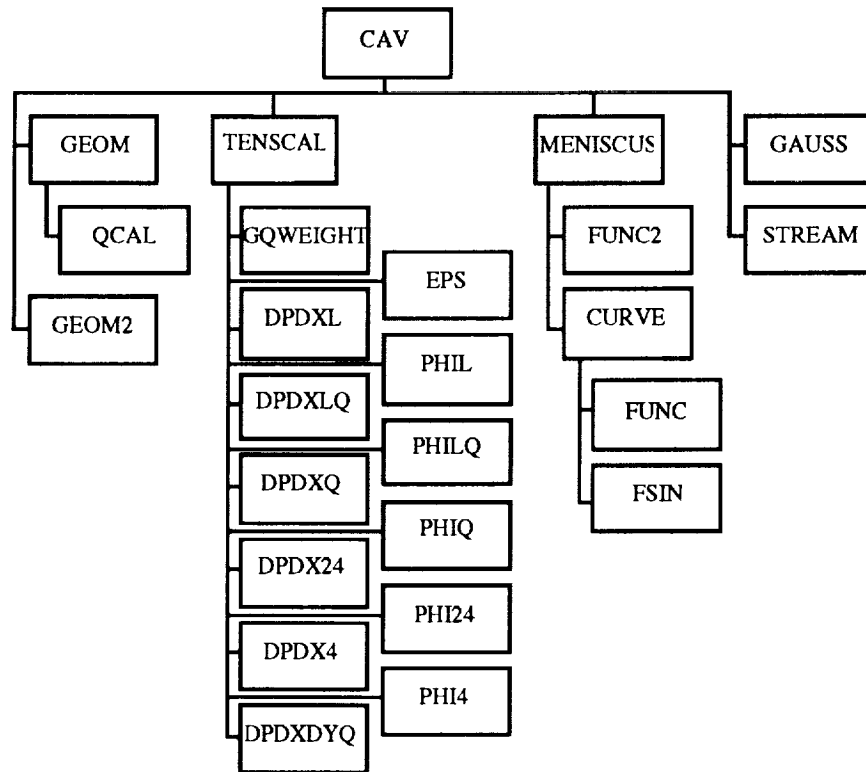
Figure B-4. CIC code structure.

geometric properties of surface elements, such as unit normal/tangent vectors and surface lengths, are also determined. This subprogram is critical because it specifies the geometric characteristics that enable calculation of finite element coefficients. The QCAL function is used several times to estimate intermediate points in the quadratic elements. Based on the x_1 and x_2 values at three known points, it calculates second-order coefficients and outputs a value of x_2 corresponding to an input value of x_1.

The GEOM2 subprogram was originally written to permit incorporation of smaller element sizes in the upper corner and sidewall regions. Using FSCALE(2) and NSMALL(2) the elements in the half-cavity along the sidewall and meniscus are contracted relative to those in the center. This subprogram was used with marginal success and did not improve accuracy significantly. Its operation is disabled by setting NSMALL = 0.

TENSCAL is called after GEOM in the CAV module and calculates the coefficient tensors used in the CAV finite element equations. In the subprogram, $(J_{ik})^{-1}$ and weighting coefficients are determined and used in several Gaussian quadrature integrations of the coefficients. These integrations refer to several subprograms representing the interpolation functions and their various gradient forms defined in appendix A. Once calculated, these coefficients are transferred back to the main program via the TENSOR common block. In addition to the coefficients applied for the domain variables, this subprogram also calculates the second-order surface tensor used in application of the velocity Neumann condition for thermocapillary flow.

Calculation of the base function value at individual nodes is performed with five different functions. For types 1 to 5, one has PHIL, PHILQ, PHIQ, PHI24, and PHI4, respectively. Calculation

of the first-order gradients of the type 1 to 5 interpolation formulas are performed in DPDXL, DPDXLQ, DPDXQ, DPDX24, and DPDX4. In all cases, $(J_{ik})^{-1}$ and the coordinate coefficients are calculated in TENSCAL and used in the determination of $\Phi_{N,i}$. The SIMPLER pressure correction equation contains two terms containing second-order gradients. These coefficients make use of the DPDXDYQ function.

TENSCAL calls out two other subprograms. GQWEIGHT is used to determine the Gaussian quadrature absissae and ordinate values used in the TENSCAL integrations. These values are permanently encoded via DATA statements in the subroutine. Another function is EPS which performs the same operation as the Einstein permutation symbol ε_{ij}.

The third main module called by CAV is the MENISCUS subroutine. In addition to controlling the iteration and convergence check on meniscus solution, it also calculates the coefficients used in the second-order representation of surface velocity and temperature, and the combined first/second-order variation of pressure. Its ancillary function FUNC2 is used to estimate coefficients for a logarithmic curve fit when Bi and the temperature gradient near the sidewall are very high.

The actual integral solution of the surface for each estimate of center line curvature is performed in the CURVE subroutine. Here, the integration begins at the center line and continues until the surface intercepts the sidewall. The resulting value of contact angle is used in MENISCUS to estimate a new center line curvature and to guide convergence to a solution. The CURVE subroutine calls two functions, FUNC and FSIN. FUNC calculates an intermediate value of temperature or pressure based on the coefficients calculated in MENISCUS. Its logic distinguishes between first, second-order, and logarithmic curve fits. FSIN is used merely to ensure a consistent definition of the contour angle in instances where the curvature is negative along the surface.

The last two subprograms called by CAV are used in specific calculation procedures. GAUSS is a Gauss-Jordan elimination routine that solves nodal pressure values given a coefficient and objective matrix. STREAM is used prior to the unit 9 data dump and calculates stream function based on the values of velocity and node coordinates.

152

APPENDIX C

CIC PROGRAM SOURCE CODE

A. Main Program Module (CAV)

```
C*********************************************************************
      PROGRAM CAV
C*********************************************************************
C Main program module of Combined Interfacial Convection (CIC) model.
C CIC calculates the steady-state velocities, stream functions,
C temperatures, pressures and surface morphology of a two-dimensional,
C rectangular liquid pore subjected to various heating conditions.
C Program employs finite element equations based on the Galerkin method.
C Vertical sides defined by solid isothermal surfaces while upper
C surface represented by deformable meniscus. Lower horizontal surface
C is open to large reservoir that permits liquid flow into and out of
C cavity. Evaporation or condensation allowed to occur depending on
C whether surfaces 1 to 3 are superheated or subcooled with respect to
C the isothermal saturated vapor above the meniscus. The main convective
C effects accounted for in the model include: thermocapillary stress
C caused by surface temperature gradients, buoyancy, and interfacial
C mass transfer.
C
C This program represents a culmination of analytical efforts in support
C of the Liquid Acquisition Device Characterization, a NASA MSFC CDDF
C research project begun in Nov 1990. Final version of program written
C by EP53/George R. Schmidt on 3/17/93.
C*********************************************************************

C Array dimension parameters

      PARAMETER (MXE= 25,         MYE= 20) ! # elements in x & y direcs
      PARAMETER (MXN= 1+2*MXE)             ! # nodes in x-direction
      PARAMETER (MYN= 1+2*MYE)             ! # nodes in y-direction
      PARAMETER (MTE= MXE*MYE)             ! total # elements
      PARAMETER (MTN= MXN*MYN)             ! total # nodes (quadratic)
      PARAMETER (MTE2= MXE-2+2*(MYE-1))    ! # 6-node pressure elements
      PARAMETER (MTE1= MTE-MTE2-2)         ! # 4-node pressure elements
      PARAMETER (MTC1= (1+MXE)*(1+MYE))    ! # corner nodes
      PARAMETER (MTC2= 2*(1+MYE)+1+MXE+2)  ! extra nodes for mixed press
      PARAMETER (MTC= MTC1+MTC2)           ! total # press nodes
      PARAMETER (MTN2= MTN+4*MYN+2*MXN+8)  ! tot # nodes (w/4th-order)

C Main program common blocks

      COMMON/VINTEG/XI(10,2),W(10)

      COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

      COMMON/VQUAD/
     *A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     *XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)
```

```
      COMMON/GRID/
     *XG(MTN2,2),VEL(MTN,2),PSI(MTN),SURFL(4,MXE),SURFN(4,MXE,2),
     *SURFT(4,MXE,2)

      COMMON/GRIDTYPE/
     *CL(MTE1,4),CLLQ(MTE2,6),CLQ(2,9),ECL(MTE,2),CQ(MTE,9),GNODE(MTC),
     *CNODE(MTN),C24(MTE2,15),C44(2,25),PNODE(36),NSURF(4),
     *SURFNODE(4,MXN),CSURF(4,MXE,3),CE(MTN2,4),NCE(MTN2),
     *CES(MXN,2),NCES(MXN),IDIRICH(MTN2)

      COMMON/SURFTYPE/ NESURF(4)

      COMMON/TENSOR/
     *EA(MTE,9,9),         EB(MTE,9,9,9,2),       EC(MTE,9,9),
     *EA24V(MTE2,9,15),    EB24(MTE2,15,9,15,2),  EC24(MTE2,15,15),
     *EA44V(2,9,25),       EB44(2,25,9,25,2),     EC44(2,25,25),
     *EA24T(MTE2,15,15),   EA44T(2,25,25),
     *ED(MTE1,9,4,2),      EDLQ(MTE2,9,6,2),      EDQ(2,9,9,2),
     *EE(MTE1,4,4),        EELQ(MTE2,6,6),        EEQ(2,9,9),
     *EF(MTE1,4,9,2),      EFLQ(MTE2,6,9,2),      EFQ(2,9,9,2),
     *EG(MTE1,4,9,2,9,2),EGLQ(MTE2,6,9,2,9,2),  EGQ(2,9,9,2,9,2),
     *EH(MTE1,4,9,9,2,2),EHLQ(MTE2,6,9,9,2,2),  EHQ(2,9,9,9,2,2),
     *EQ(MTE,9,2,9,2),     SAQ(MXE,3,9),          SAQ2(2,5,25)

      COMMON/MENVAR/
     * ADUM(3,MXE),BDUM(3,MXE),CDUM(3,MXE),APDUM(2),AQDUM(2)

      COMMON/SURFACE/
     *XSURF(MXN),YSURF(MXN),YSURF0(MXN),PSURF(MXN),PBUB(MXN),
     *TSURF(MXN),SURFNN(MXN,2),SURFNT(MXN,2)

      COMMON/PLOT/
     *IMODE,IOUT8,IOUT9

      COMMON/MXTRAN/
     >AMAT(MTC,MTC),BVEC(MTC),XVEC(MTC)

C Specification of other arrays and program variable types

      DIMENSION YSTAT(MXN)
      DIMENSION TEMP(MTN2),TEMP0(MTN2),PREV(MTN2,2)
      DIMENSION PDYN(MTN),PDYN0(MTN)
      DIMENSION VEL0(MTN,2),DVEL(MTN,2)
      DIMENSION DP(MTC)
      DIMENSION ACCEL(2),NSN(3),NSMALL(2),FSCALE(2)
      DIMENSION SB(3,3),SC(3),SB2(5,5),SC2(5)

      DOUBLE PRECISION ADUM,BDUM,CDUM,DDUM,XSURF,YSURF,ALPHA,PSURF,TSURF
      DOUBLE PRECISION SURFNN,SURFNT,YSURF0,PBUB,APDUM,AQDUM,DVREC
      DOUBLE PRECISION DCANG,DBO,DCA,DMA,DPR,DREVAP,DPREF,DRASPECT
      DOUBLE PRECISION DSUM,YERR,DPVAP,MENTOL,DYREF,DBI,DCR,DFRHO
      DOUBLE PRECISION X1,X2,X3,V1,V2,V3,AC,BC,DER
      INTEGER CL,CQ,CLLQ,CLQ,ECL,GNODE,CNODE,PNODE,SURFNODE,CSURF
      INTEGER GAM,BETA,R,S,P,CE,CES,C24,C44
      REAL MA,MNM1,MN3,MN4,JINV,LHS,MAIN
      CHARACTER*12 TITF,TITF2,QUETYPE
      CHARACTER*80 TIT(3)
      CHARACTER*40 TITLINE
```

```fortran
C Initialization of non-input parameters

      DATA NLNQ,NLNL/9,4/  ! Number of local nodes (quad and linear)
      DATA TPARAM/.5/      ! Time step factor (Crank-Nicholson)
      DATA ACCEL/0.,1./    ! Normalized acceleration
      IMODE= 2             ! 0= VAX batch, 1= VAX inter, 2= CRAY batch
      IGO= 1               ! Stop execution flag- used in CRAY batch mode

C Integral tensors used in formulating meniscus surface contributions

      DATA (SB(1,I),I=1,3)/  4.,   2.,  -1./,
     *     (SB(2,I),I=1,3)/  2.,  16.,   2./,
     *     (SB(3,I),I=1,3)/ -1.,   2.,   4./

      DATA (SB2(1,I),I=1,5)/ 292., 296.,-174.,  56.,  -29./,
     *     (SB2(2,I),I=1,5)/ 296.,1792.,-384., 256.,   56./,
     *     (SB2(3,I),I=1,5)/-174.,-384.,1872.,-384., -174./,
     *     (SB2(4,I),I=1,5)/  56., 256.,-384.,1792.,  296./,
     *     (SB2(5,I),I=1,5)/ -29.,  56.,-174., 296.,  292./

      DATA SC / 1.,   4.,   1./
      DATA SC2/ 7.,  32.,  12.,  32.,   7./

C Input/output device assignment. Data input and read statements

      IF (IMODE.EQ.2) THEN
        OPEN(UNIT=7,FILE='input.dat',STATUS='UNKNOWN')
        TITF= 'input.dat    '
      ELSE IF (IMODE.EQ.1) THEN
        WRITE(6,1000)
        READ(5,1040)TITF
        OPEN(UNIT=7,FILE=TITF,STATUS='OLD')
      ELSE
        TITF= 'VAX.BATCH
      END IF

      READ (7,1002)(TIT(J),J=1,3)
      READ (7,1070)QUETYPE,TITF2
      READ (7,1003)TITLINE,IOUT8
      READ (7,1003)TITLINE,IOUT9

      IF (IMODE.EQ.2) THEN
         IF (IOUT8.NE.0) OPEN(UNIT=8,FILE='output.dat',STATUS='UNKNOWN')
         IF (IOUT9.NE.0) OPEN(UNIT=9,FILE='plot.dat',STATUS='UNKNOWN')
      ELSE IF (IMODE.EQ.1) THEN
         IF (IOUT8.NE.0) OPEN(UNIT=8,FILE='PORE.OUT',STATUS='NEW')
         IF (IOUT9.NE.0) OPEN(UNIT=9,FILE='PORE.PLOT',STATUS='NEW')
      END IF

      IF (IOUT8.NE.0) WRITE(8,1002)(TIT(J),J=1,3)
      READ (7,1003)TITLINE,ITYPE
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,ITYPE
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,ITYPE
      READ (7,1003)TITLINE,IPRESS
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,IPRESS
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,IPRESS
      READ (7,1003)TITLINE,ICURVE
```

155

```
IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,ICURVE
IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,ICURVE
READ (7,1003)TITLINE,ISYMM
IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,ISYMM
IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,ISYMM
READ (7,1003)TITLINE,ISSTEMP
IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,ISSTEMP
IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,ISSTEMP

READ(7,1070)TITF,TITF2
READ (7,1005)TITLINE,CANG
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,CANG
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,CANG
READ (7,1005)TITLINE,BO
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,BO
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,BO
READ (7,1005)TITLINE,CA
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,CA
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,CA
READ (7,1005)TITLINE,MAIN
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,MAIN
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,MAIN
READ (7,1005)TITLINE,RVOLSURF
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,RVOLSURF
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,RVOLSURF
READ (7,1005)TITLINE,PR
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,PR
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,PR
READ (7,1005)TITLINE,REVAPIN
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,REVAPIN
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,REVAPIN
READ (7,1005)TITLINE,CR
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,CR
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,CR
READ (7,1005)TITLINE,FRHO
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,FRHO
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,FRHO
READ (7,1005)TITLINE,LHS
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,LHS
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,LHS
READ (7,1005)TITLINE,BI
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,BI
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,BI
READ (7,1005)TITLINE,PCORNER
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,PCORNER
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,PCORNER
READ (7,1005)TITLINE,TINIT
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,TINIT
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,TINIT
READ (7,1005)TITLINE,RASPECT
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,RASPECT
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,RASPECT

READ(7,1070)TITF,TITF2
READ (7,1005)TITLINE,VTOL
IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,VTOL
IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,VTOL
READ (7,1005)TITLINE,PTOL
```

```
      IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,PTOL
      IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,PTOL
      READ (7,1005)TITLINE,TTOL
      IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,TTOL
      IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,TTOL
      READ (7,1005)TITLINE,SSTOL
      IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,SSTOL
      IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,SSTOL
      READ (7,1005)TITLINE,MENTOL
      IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,MENTOL
      IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,MENTOL
      READ (7,1005)TITLINE,DELTMAX
      IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,DELTMAX
      IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,DELTMAX

      READ(7,1070)TITF,TITF2
      READ (7,1003)TITLINE,NTSTEP
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NTSTEP
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NTSTEP

      READ (7,1003)TITLINE,NXE
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NXE
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NXE
      READ (7,1003)TITLINE,NSMALL(1)
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NSMALL(1)
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NSMALL(1)
      READ (7,1005)TITLINE,FSCALE(1)
      IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,FSCALE(1)
      IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,FSCALE(1)

      READ (7,1003)TITLINE,NYE
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NYE
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NYE
      READ (7,1003)TITLINE,NSMALL(2)
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NSMALL(2)
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NSMALL(2)
      READ (7,1005)TITLINE,FSCALE(2)
      IF (IOUT8.NE.0) WRITE(8,1005)TITLINE,FSCALE(2)
      IF (IOUT9.NE.0) WRITE(9,1074)TITLINE,FSCALE(2)

      READ (7,1003)TITLINE,NPOINT
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NPOINT
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NPOINT
      READ (7,1003)TITLINE,NINT
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NINT
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NINT
      READ (7,1003)TITLINE,NMA
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NMA
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NMA
      READ (7,1003)TITLINE,NREVAP
      IF (IOUT8.NE.0) WRITE(8,1003)TITLINE,NREVAP
      IF (IOUT9.NE.0) WRITE(9,1073)TITLINE,NREVAP

      READ(7,1070)TITF,TITF2
      DO 1 N=1,6
      READ (7,1071)TITLINE,(PNODE((N-1)*6+K),K= 1,6)
      IF (IOUT8.NE.0) WRITE(8,1071)TITLINE,(PNODE((N-1)*6+K),K= 1,6)
    1 CONTINUE
```

```
      IF (IOUT9.GT.0) THEN
      WRITE(9,1072)
      DO 2 I=1,3
        READ(7,1003)TITLINE,NCONTOUR
        WRITE(9,1073)TITLINE,NCONTOUR
    2   CONTINUE
      DO 3 I=1,3
        READ(7,1005)TITLINE,XARROW
        WRITE(9,1074)TITLINE,XARROW
    3   CONTINUE
      TITLINE=' Number of top nodes
      WRITE(9,1073)TITLINE,2*NXE+1
      TITLINE=' Number of global nodes
      WRITE(9,1073)TITLINE,(2*NXE+1)*(2*NYE+1)
      END IF

      FT0= 1.-TPARAM        ! Temporal param factor (prev time step)
      FT1= TPARAM           ! Temporal param factor (current time step)
      RA= BO*MAIN*RVOLSURF  ! Rayleigh number
      IF (MAIN.NE.0.) CA= CR*PR/MAIN
      VREC= CA*(FRHO-1.)/REVAPIN/REVAPIN

      IF (IPRESS.EQ.0) THEN
        NLNS= 4
        NLNC= 4
      ELSE IF (IPRESS.EQ.1) THEN
        NLNS= 6
        NLNC= 9
      END IF

      IF (ITYPE.EQ.0) THEN
        TVAP= 0.
      ELSE IF (ITYPE.EQ.1) THEN
        TVAP= 1.
      END IF

C Marangoni and evap resistance step value

      DELMA= MAIN/REAL(NMA)
      DELRINV= 1./REVAPIN/REAL(NREVAP)

C Double precision representations of thermophysical parameters-input
C for meniscus calculation

      DCANG= DBLE(CANG)
      DBO= DBLE(BO)
      DCA= DBLE(CA)
      DMA= DBLE(MAIN)
      DPR= DBLE(PR)
      DBI= DBLE(BI)
      DREVAP= DBLE(REVAPIN)
      DRASPECT= DBLE(RASPECT)
      DYREF= DBLE(RASPECT)
      DCR= DBLE(CR)
      DFRHO= DBLE(FRHO)
      DVREC= DBLE(VREC)

C Number of nodes, elements and velocity values
```

```
            NXP= 2*NXE+1          ! # global nodes in X1-direction
            NYP= 2*NYE+1          ! # global nodes in X2-direction
            NGNODE= NXP*NYP       ! # global nodes (quad var)
            NELEMENT= NXE*NYE     ! # elements
            NROW= 2*NXP           ! # global nodes in X1 row of elements
            NCOL= 2*NYP           ! # global nodes in X2 col of elements
            NVEL= 2*NGNODE        ! # velocity values
            NGNODE2= NGNODE+4*NYP+2*NXP+8  ! # nodes (4th-order var)

C Initial reference for surface geometry and conditions

            DO 4 K= 1,NXP
              PSURF(K)= 0.
              PBUB(K)= 0.
              TSURF(K)= TVAP
              YSURF0(K)= DRASPECT
          4 CONTINUE

C Variable grid geometry control

            CALL GEOM2(NXE,NYE,NXP,NROW,NSMALL,RASPECT,FSCALE)

C Static, isothermal meniscus geometry (T= 0)

            CALL MENISCUS(DCANG,DBO,DCA,DMA,DPR,DBI,DREVAP,DPREF,DRASPECT,
          >      DPVAP,DYREF,DCR,DFRHO,DVREC,NINT,NXP,ISYMM,ICURVE,ITYPE,
          >      IPRESS)
            PREF= REAL(DPREF)

            IF (IMODE.EQ.1) THEN
              WRITE(6,1073) ' STATIC MENISCUS GEOMETRY',0
              WRITE(6,1007) 0
            END IF
            DO 5 K= 1,NXP
              IF (IMODE.EQ.1) WRITE(6,1008)K,XSURF(K),YSURF(K),
          >        SURFNN(K,1),SURFNN(K,2),SURFNT(K,1),SURFNT(K,2)
              YSTAT(K)= YSURF(K)
              YSURF0(K)= YSURF(K)
          5 CONTINUE

            IF (ISSTEMP.EQ.1) THEN

            CALL GEOM(NXP,NYP,NXE,NYE,NGNODE,NELEMENT,NCORNER,NROW,NCOL,
          >        IPRESS,NGNODE2,RASPECT,DXE,DYE,DXN,DYN,DXMIN,DYMIN,VOL)

            CALL TENSCAL(NELEMENT,NLNQ,NLNL,NPOINT,IPRESS,PR,PVAP)

            DO 10 NG= 1,NGNODE2
              IF (IDIRICH(NG).EQ.0) THEN
                PREV(NG,1)= 1.-TVAP
              ELSE
                TEMP(NG)= 1.-TVAP
                PREV(NG,1)= TEMP(NG)
              END IF
         10 CONTINUE

            DO 20 ITTEMP= 1,1000000
```

```fortran
      DO 11 NG= 1,NGNODE2
        IF (IDIRICH(NG).EQ.1) GOTO 11
        AMAT0= 0.
        BVEC0= 0.

        DO 14 NNE= 1,NCE(NG)

          NE= CE(NG,NNE)
          IF (ECL(NE,1).EQ.0) THEN
            NP= NLNQ
          ELSE IF (ECL(NE,1).EQ.1) THEN
            NP= 15
          ELSE IF (ECL(NE,1).EQ.2) THEN
            NP= 25
          END IF
          DO 12 N= 1,NP
            IF (ECL(NE,1).EQ.0) THEN
              IF (CQ(NE,N).EQ.NG) GOTO 13
            ELSE IF (ECL(NE,1).EQ.1) THEN
              IF (C24(ECL(NE,2),N).EQ.NG) GOTO 13
            ELSE IF (ECL(NE,1).EQ.2) THEN
              IF (C44(ECL(NE,2),N).EQ.NG) GOTO 13
            END IF
12        CONTINUE
          STOP
13        CONTINUE

          DO 14 M= 1,NP
            IF (ECL(NE,1).EQ.0) THEN
            MG= CQ(NE,M)
            IF (MG.EQ.NG) THEN
              AMAT0= AMAT0+EC(NE,N,M)/PR
            ELSE
              BVEC0= BVEC0-EC(NE,N,M)/PR*PREV(MG,1)
            END IF
            ELSE IF (ECL(NE,1).EQ.1) THEN
            MG= C24(ECL(NE,2),M)
            IF (MG.EQ.NG) THEN
              AMAT0= AMAT0+EC24(ECL(NE,2),N,M)/PR
            ELSE
              BVEC0= BVEC0-EC24(ECL(NE,2),N,M)/PR*PREV(MG,1)
            END IF
            ELSE IF (ECL(NE,1).EQ.2) THEN
            MG= C44(ECL(NE,2),M)
            IF (MG.EQ.NG) THEN
              AMAT0= AMAT0+EC44(ECL(NE,2),N,M)/PR
            ELSE
              BVEC0= BVEC0-EC44(ECL(NE,2),N,M)/PR*PREV(MG,1)
            END IF
            END IF
14        CONTINUE

C TEMPERATURE BOUNDARY CONDITIONS

C MIXED TEMPERATURE BC FOR SURFACE 4

      IF ((NG.LE.NXP).OR.(NG.GE.NGNODE+1.AND.NG.LE.NGNODE+4)) THEN
```

```
              IF (NG.LE.NXP) THEN
                NP= NCES(NG)
              ELSE
                NP= 1
              END IF
           DO 15 NNE= 1,NP

              IF (NG.LE.NXP) THEN
              NE= CES(NG,NNE)
              ELSE IF (NG.LE.NGNODE+2) THEN
                NE= 1
              ELSE
                NE= NESURF(4)
              END IF

              IF (NE.EQ.1.OR.NE.EQ.NESURF(4)) THEN
                DO 18 N= 1,5
18                 IF (C44(ECL(NE,2),14-N).EQ.NG) GOTO 19
                STOP
19              CONTINUE
                DCOEFF= -1.*BI/PR*SURFL(4,NE)/5670.
                NM= 5
                IF (ITYPE.EQ.1)
      >           BVEC0= BVEC0+BI/PR*SURFL(4,NE)*SC2(N)/90.
              ELSE
                DO 16 N= 1,3
16                 IF (CSURF(4,NE,N).EQ.NG) GOTO 17
                STOP
17              CONTINUE
                DCOEFF= -1.*BI/PR*SURFL(4,NE)/30.
                NM= 3
                IF (ITYPE.EQ.1)
      >           BVEC0= BVEC0+BI/PR*SURFL(4,NE)*SC(N)/6.
              END IF

              DO 15 M= 1,NM
                IF (NE.EQ.1.OR.NE.EQ.NESURF(4)) THEN
                  MG= C44(ECL(NE,2),14-M)
                  IF (MG.NE.1.AND.MG.NE.NXP) THEN
                  IF (MG.EQ.NG) THEN
                    AMAT0= AMAT0-DCOEFF*SB2(N,M)
                  ELSE
                    BVEC0= BVEC0+DCOEFF*SB2(N,M)*PREV(MG,1)
                  END IF
                  END IF
                ELSE
                  MG= CSURF(4,NE,M)
                  IF (MG.EQ.NG) THEN
                    AMAT0= AMAT0-DCOEFF*SB(N,M)
                  ELSE
                    BVEC0= BVEC0+DCOEFF*SB(N,M)*PREV(MG,1)
                  END IF
                END IF
15            CONTINUE
           END IF
           TEMP(NG)= .3*(BVEC0/AMAT0-PREV(NG,1))+PREV(NG,1)
```

```
   11 CONTINUE

      TOTV= 0.
      VERR= 0.
      DO 9318 NG= 1,NGNODE2
      TOTV= TOTV+PREV(NG,1)*PREV(NG,1)
      ERR= TEMP(NG)-PREV(NG,1)
      VERR= VERR+ERR*ERR
      PREV(NG,1)= TEMP(NG)
 9318 CONTINUE

      IF (TOTV.NE.0.) THEN
      FERR= SQRT(VERR/TOTV)
      ELSE
      FERR= 1.
      END IF

      IF (IMODE.EQ.1) THEN
      WRITE(6,1083)FERR
 1083 FORMAT(' TEMP ERROR= ',E12.6)
      WRITE(6,1025)
      WRITE(6,1022)(TEMP(PNODE(N)),N=1,36)
      END IF

C Intermittent update and continuation check

      IF (IMODE.EQ.2) THEN

      IF (QUETYPE.EQ.'HUGE3') THEN
        OPEN(UNIT=2,FILE='huge3.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='huge3.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'HUGE2') THEN
        OPEN(UNIT=2,FILE='huge2.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='huge2.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE4') THEN
        OPEN(UNIT=2,FILE='large4.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='large4.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE3') THEN
        OPEN(UNIT=2,FILE='large3.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='large3.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE2') THEN
        OPEN(UNIT=2,FILE='large2.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='large2.dat',STATUS='UNKNOWN')
      ELSE
        OPEN(UNIT=2,FILE='other.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='other.dat',STATUS='UNKNOWN')
      END IF

      READ(2,1003) TITLINE,IGO
      CLOSE(2)

        WRITE(3,3919)ITMEN,1.
        WRITE(3,1030)ITTEMP,TIME
        WRITE(3,6391)VSSCON,PSSCON,FERR
        WRITE(3,1025)
        WRITE(3,1022)(TEMP(PNODE(N)),N= 1,36)
        CLOSE(3)
```

```
      IF (IGO.EQ.0) THEN
        IF (QUETYPE.EQ.'HUGE3') THEN
          OPEN(UNIT=2,FILE='huge3.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'HUGE2') THEN
          OPEN(UNIT=2,FILE='huge2.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'LARGE4') THEN
          OPEN(UNIT=2,FILE='large4.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'LARGE3') THEN
          OPEN(UNIT=2,FILE='large3.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'LARGE2') THEN
          OPEN(UNIT=2,FILE='large2.tst',STATUS='UNKNOWN')
        ELSE
          OPEN(UNIT=2,FILE='other.tst',STATUS='UNKNOWN')
        END IF
        WRITE(2,1003) TITLINE,1
        CLOSE(2)
          GOTO 503
      END IF

      END IF

      IF (FERR.LE.TTOL) GOTO 30

   20 CONTINUE
   30 CONTINUE

      DO 9319 NG= 1,NGNODE2
 9319 TEMP0(NG)= TEMP(NG)

      END IF

      NB= NGNODE-NXP
      DO 21 K= 1,NXP
      IF (ISSTEMP.EQ.1) THEN
        TSURF(K)= TEMP(K)
      ELSE
        TSURF(K)= DBLE(TINIT)
      END IF
   21 CONTINUE

C********************************************************************
C MENISCUS GEOMETRY LOOP
C********************************************************************

      MA= 0.
      RINV= 0.

      DO 500 ITMEN= 1,100

      CALL MENISCUS(DCANG,DBO,DCA,DMA,DPR,DBI,DREVAP,DPREF,DRASPECT,
     >     DPVAP,DYREF,DCR,DFRHO,DVREC,NINT,NXP,ISYMM,ICURVE,ITYPE,
     >     IPRESS)
      PREF= REAL(DPREF)

      IF (IOUT8.GE.2) WRITE(8,1007) ITMEN
      IF (IMODE.EQ.1) WRITE(6,1007) ITMEN
      DO 501 K= 1,NXP
        IF (IOUT8.GE.2) WRITE(8,1008)K,XSURF(K),YSURF(K),
```

```
   >       SURFNN(K,1),SURFNN(K,2),SURFNT(K,1),SURFNT(K,2)
          IF (IMODE.EQ.1) WRITE(6,1008)K,XSURF(K),YSURF(K),
   >       SURFNN(K,1),SURFNN(K,2),SURFNT(K,1),SURFNT(K,2)
  501 CONTINUE

      DSUM= 0.
      YERR= 0.
      DO 502 K= 1,NXP,2
        DSUM= DSUM+YSURFO(K)*YSURFO(K)
      YERR= YERR+(YSURF(K)-YSURFO(K))*(YSURF(K)-YSURFO(K))
  502 CONTINUE
      YERR= DSQRT(YERR/DSUM)
      IF (YERR.LE.MENTOL.AND.ITMEN.GT.1) THEN
      IF (IOUT9.EQ.1) THEN
        TITLINE=' MENISCUS DIM CHARACTERISTICS'
        WRITE(9,1073)TITLINE,0
        DO 603 K= 1,NXP
          WRITE(9,1009)XSURF(K),YSTAT(K),YSURF(K),SURFNN(K,1),
   >                    SURFNN(K,2),SURFNT(K,1),SURFNT(K,2)
  603      CONTINUE
      END IF
      GOTO 503
      END IF

      DO 509 K= 1,NXP
  509 YSURF(K)= .5*(YSURF(K)+YSURFO(K))
      DO 521 K= 1,NXP
        IF (K.EQ.1) THEN
          KREF= 2
        ELSE IF (K.EQ.NXP) THEN
          KREF= NXP-1
        ELSE
          KREF= K
        END IF
        V1= YSURF(KREF-1)
        V2= YSURF(KREF)
        V3= YSURF(KREF+1)
        X1= XSURF(KREF-1)
        X2= XSURF(KREF)
        X3= XSURF(KREF+1)
        AC= ((V1-V2)/(X1-X2)-(V2-V3)/(X2-X3))/(X1-X3)
        BC= (V1-V2)/(X1-X2)-AC*(X1+X2)
        DER= 2.*AC*XSURF(K)+BC
        SURFNT(K,1)= 1./DSQRT(DER**2+1.)
        SURFNT(K,2)= DER*SURFNT(K,1)
        SURFNN(K,1)= -1.*SURFNT(K,2)
        SURFNN(K,2)= SURFNT(K,1)
      YSURFO(K)= YSURF(K)
  521 CONTINUE

C*******************************************************************
C OVERALL DOMAIN GEOMETRY
C*******************************************************************

      CALL GEOM(NXP,NYP,NXE,NYE,NGNODE,NELEMENT,NCORNER,NROW,NCOL,
   >      IPRESS,NGNODE2,RASPECT,DXE,DYE,DXN,DYN,DXMIN,DYMIN,VOL)

C*******************************************************************
```

```
C ELEMENT TENSORS
C***********************************************************************

        CALL TENSCAL(NELEMENT,NLNQ,NLNL,NPOINT,IPRESS,PR,PVAP)

C***********************************************************************
C DEPENDENT VARIABLE INITIALIZATION
C***********************************************************************

C Velocities and pressures. Set equal to zero at all points.

        IFREEZE= 1
        DELT= DELTMAX
        TSSCON= 1.
        DO 504 N= 1,NGNODE
          PDYN0(N)= PCORNER
          PDYN(N)= PDYN0(N)
          PSI(N)= 0.
          DO 504 K= 1,2
            VEL0(N,K)= 0.
            VEL(N,K)= VEL0(N,K)
    504 CONTINUE
        IF (ISSTEMP.NE.1) THEN
          DO 506 N= 1,NGNODE2
            TEMP0(N)= TINIT
            TEMP(N)= TEMP0(N)
    506     CONTINUE
        ELSE
          DO 507 N= 1,NGNODE2
            TEMP0(N)= TEMP(N)
    507     CONTINUE
        END IF

        IF (IMODE.EQ.1) THEN
          WRITE(6,1020)
          WRITE(6,1021)
          WRITE(6,1022)(VEL(PNODE(N),1),N= 1,36)
          WRITE(6,1023)
          WRITE(6,1022)(VEL(PNODE(N),2),N= 1,36)
          WRITE(6,1024)
          WRITE(6,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
          WRITE(6,1025)
          WRITE(6,1022)(TEMP(PNODE(N)),N= 1,36)
        END IF
        IF (IOUT8.GE.2) THEN
          WRITE(8,1020)
          WRITE(8,1021)
          WRITE(8,1022)(VEL(PNODE(N),1),N= 1,36)
          WRITE(8,1023)
          WRITE(8,1022)(VEL(PNODE(N),2),N= 1,36)
          WRITE(8,1024)
          WRITE(8,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
          WRITE(8,1025)
          WRITE(8,1022)(TEMP(PNODE(N)),N= 1,36)
        END IF

C***********************************************************************
C***********************************************************************
```

```
C TIME INTERVAL/STEPPING LOOP
C**********************************************************************
C**********************************************************************

        TIME= 0.
        ICONV= 0
        MA= 0.
        RINV= 0.
        ITCONV= 0
        DO 900 ITIME= 1,NTSTEP

        IF (ITIME.NE.2) THEN
          IF (MA.LT.MAIN) THEN
            MA= MA+DELMA
          ELSE IF (MA.GE.MAIN) THEN
            MA= MAIN
          END IF
          IF (RINV.LT.1./REVAPIN) THEN
            RINV= RINV+DELRINV
          ELSE IF (1./REVAP.GE.1./REVAPIN) THEN
            RINV= 1./REVAPIN
          END IF
        REVAP= 1./RINV
        END IF

        IF (IMODE.EQ.1) WRITE(6,1038)ITIME

C**********************************************************************
C**********************************************************************
C TEMPERATURE ITERATION/CONVERGENCE LOOP
C**********************************************************************
C**********************************************************************

        DO 100 ITTEMP= 1,10000

        IF (IMODE.EQ.1) WRITE(6,1042)ITTEMP
        IF (ITIME.EQ.1.AND.ITTEMP.EQ.1) GOTO 101

C Volumetric flowrate through upper surface due to evaporation

        DXMIN= XG(NGNODE,1)-XG(NGNODE-1,1)
        QFLOW= 0.
        DO 102 NE= 1,NESURF(4)
        SUM= 0.
        IF (NE.EQ.1.OR.NE.EQ.NESURF(4)) THEN
          NM= 5
        ELSE
          NM= 3
        END IF
        DO 103 N= 1,NM
          IF (NM.EQ.5) THEN
            IF (NE.EQ.1.AND.N.EQ.1) GOTO 103
            IF (NE.EQ.NESURF(4).AND.N.EQ.5) GOTO 103
            SUM= SUM+SC2(N)*TEMP(C44(ECL(NE,2),14-N))
          ELSE IF (NM.EQ.3) THEN
            IF (CSURF(4,NE,N).EQ.1)GOTO 103
            IF (CSURF(4,NE,N).EQ.NXP)GOTO 103
            SUM= SUM+SC(N)*TEMP(CSURF(4,NE,N))
```

```
              END IF
 103     CONTINUE
         IF (NM.EQ.3) THEN
            QFLOW= QFLOW+SURFL(4,NE)/6.*RINV*SUM
         ELSE
            QFLOW= QFLOW+SURFL(4,NE)/90.*RINV*SUM
         END IF
         IF (ITYPE.EQ.1) QFLOW= QFLOW-SURFL(4,NE)*RINV
 102  CONTINUE
      VELBOT= 3.*QFLOW/(3.-DXMIN)

C*********************************************************************
C*********************************************************************
C PRESSURE CORRECTION ITERATION/CONVERGENCE LOOP
C*********************************************************************
C*********************************************************************

      DO 200 ITPRESS= 1,1000

      IF (IMODE.EQ.1) WRITE(6,1031)ITPRESS

C*********************************************************************
C*********************************************************************
C VELOCITY/PRESSURE ITERATION/CALCULATION LOOP
C*********************************************************************
C*********************************************************************

      DO 300 ITVEL= 1,1000

      IF (IMODE.EQ.1) WRITE(6,1032)ITVEL

C Time step check and adjustment

      V1MAX= VEL(1,1)
      V2MAX= VEL(1,2)
      DO 302 N= 1,NGNODE
      IF (ABS(VEL(N,1)).GT.V1MAX) V1MAX= ABS(VEL(N,1))
      IF (ABS(VEL(N,2)).GT.V2MAX) V2MAX= ABS(VEL(N,2))
      DO 302 I= 1,2
         PREV(N,I)= VEL(N,I)
 302  CONTINUE
      IF (V1MAX.LE.1.0E-6.AND.V2MAX.LE.1.0E-6) GOTO 303
      DELTLIM= 1./(V2MAX/DYMIN+V1MAX/DXMIN)
      IF (DELTLIM.LT.DELT) DELT= DELTLIM/10.
      IF (IFREEZE.EQ.1) DELT= DELTLIM
 303  CONTINUE
      IF (IMODE.EQ.1)WRITE(6,3927)DELT
 3927 FORMAT(' DELT= ',E12.6)

C X1 and X2 component velocities solved sequentially

      DO 310 I= 1,2
      DO 310 NG= 1,NGNODE

      AMAT0= 0.
      BVEC0= 0.

      IF (NG.GE.2.AND.NG.LE.NXP-1.AND.I.EQ.2) THEN
```

```fortran
      XN1= REAL(SURFNN(NG,1))
      XN2= REAL(SURFNN(NG,2))
      IF (ITYPE.EQ.0) THEN
        FTEMP= TEMP(NG)
      ELSE
        FTEMP= TEMP(NG)-1.
      END IF
      VEL(NG,I)= (FTEMP*RINV-VEL(NG,1)*XN1)/XN2
      PREV(NG,I)= VEL(NG,I)
      GOTO 310
      END IF
      IF (IDIRICH(NG).EQ.1) THEN
      IF (I.EQ.2) THEN
        IF (NG.GE.NGNODE-NXP+2.AND.NG.LE.NGNODE-1) THEN
          VEL(NG,I)= VELBOT
        ELSE
          VEL(NG,I)= 0.
        END IF
      ELSE
        VEL(NG,I)= 0.
      END IF
      PREV(NG,I)= VEL(NG,I)
      GOTO 310
      END IF

C Summation of contributions from element to global domain

      DO 320 NNE= 1,NCE(NG)

      NE= CE(NG,NNE)
      DO 301 N= 1,NLNQ
  301 IF (CQ(NE,N).EQ.NG) GOTO 305
      STOP
  305 CONTINUE

      VRESID= 0.
      DO 321 M= 1,NLNQ
      MNM1= 0.
      DO 322 GAM= 1,NLNQ
      DO 322 J= 1,2
  322   MNM1=MNM1+EB(NE,N,GAM,M,J)*VEL0(CQ(NE,GAM),J)
  321 VRESID= VRESID+VEL0(CQ(NE,M),I)*
     *  (EA(NE,N,M)-DELT*FT0*(MNM1+EC(NE,N,M)))

      QTERM= 0.
      DO 323 M= 1,NLNQ
      DO 323 J= 1,2
      VAVG= FT0*VEL0(CQ(NE,M),J)+FT1*PREV(CQ(NE,M),J)
      QTERM= QTERM+EQ(NE,N,J,M,I)*VAVG
  323 CONTINUE

      MN3= 0.
      IF (ECL(NE,1).EQ.0) THEN
      DO 324 M= 1,NLNL
  324   MN3=MN3+ED(ECL(NE,2),N,M,I)*PDYN(CL(ECL(NE,2),M))
      ELSE IF (ECL(NE,1).EQ.1) THEN
      DO 350 M= 1,NLNS
  350   MN3=MN3+EDLQ(ECL(NE,2),N,M,I)*PDYN(CLLQ(ECL(NE,2),M))
```

```
      ELSE IF (ECL(NE,1).EQ.2) THEN
      DO 351 M= 1,NLNC
351   MN3=MN3+EDQ(ECL(NE,2),N,M,I)*PDYN(CLQ(ECL(NE,2),M))
      END IF


      MN4= 0.
      IF (ECL(NE,1).EQ.0) THEN
      DO 325 M= 1,9
      TAVG= FT0*TEMP0(CQ(NE,M))+FT1*TEMP(CQ(NE,M))
325   MN4=MN4+ RA/PR*ACCEL(I)*EA(NE,N,M)*TAVG
      ELSE IF (ECL(NE,1).EQ.1) THEN
      DO 327 M= 1,15
      TAVG= FT0*TEMP0(C24(ECL(NE,2),M))+FT1*TEMP(C24(ECL(NE,2),M))
327   MN4=MN4+ RA/PR*ACCEL(I)*EA24V(ECL(NE,2),N,M)*TAVG
      ELSE IF (ECL(NE,1).EQ.2) THEN
      DO 328 M= 1,25
      TAVG= FT0*TEMP0(C44(ECL(NE,2),M))+FT1*TEMP(C44(ECL(NE,2),M))
328   MN4=MN4+ RA/PR*ACCEL(I)*EA44V(ECL(NE,2),N,M)*TAVG
      END IF


      BVEC0= BVEC0+VRESID- DELT*(QTERM+MN3-MN4)


      DO 320 M= 1,NLNQ


      MG= CQ(NE,M)


      MNM1= 0.
      DO 326 GAM= 1,NLNQ
      DO 326 J= 1,2
326   MNM1=MNM1+EB(NE,N,GAM,M,J)*PREV(CQ(NE,GAM),J)


      IF (MG.EQ.NG) THEN
      AMAT0= AMAT0+
     *    EA(NE,N,M)+FT1*DELT*(MNM1+EC(NE,N,M))
      ELSE
      BVEC0= BVEC0-(EA(NE,N,M)+FT1*DELT*(MNM1+EC(NE,N,M)))*
     *    PREV(MG,I)
      END IF

320 CONTINUE

C SURFACE 4 NEUMAN CONDITION FOR X1-COMPONENT VELOCITY
C Marangoni effect applied on upper free surface, i.e., tangential
C stress proportional to temperature gradient.

      IF (NG.LE.NXP.AND.I.EQ.1) THEN
      DO 330 NNE= 1,NCES(NG)
      NE= CES(NG,NNE)
      DCOEFF= DELT*SURFL(4,NE)*MA/PR          ! Vol tens form
      IF (NE.EQ.1.OR.NE.EQ.NESURF(4)) THEN
        DO 334 N= 1,5
334       IF (C44(ECL(NE,2),14-N).EQ.NG) GOTO 335
        STOP
335     CONTINUE
        SM= 0.
        DO 336 M= 1,25
336       SM= SM+SAQ2(ECL(NE,2),N,M)*TEMP(C44(ECL(NE,2),M))
      ELSE
```

```fortran
         DO 331 N= 1,3
331         IF (CSURF(4,NE,N).EQ.NG) GOTO 332
         STOP
332         CONTINUE
         SM= 0.
         DO 333 M= 1,NLNQ
333         SM= SM+SAQ(NE,N,M)*TEMP(CQ(NE,M))
      END IF
      BVEC0= BVEC0-DCOEFF*SM
330 CONTINUE
      END IF

      VEL(NG,I)= .5*(BVEC0/AMAT0-PREV(NG,I))+PREV(NG,I)

310 CONTINUE

      VRMSTOT= 0.
      VRMSERR= 0.
      DO 340 NG= 1,NGNODE
      DO 340 I= 1,2
         VRMSTOT= VRMSTOT+PREV(NG,I)*PREV(NG,I)
         ERR= VEL(NG,I)-PREV(NG,I)
         VRMSERR= VRMSERR+ERR*ERR
      PREV(NG,I)= VEL(NG,I)
340 CONTINUE

      IF (VRMSTOT.NE.0.) THEN
         FVRMS= SQRT(VRMSERR/VRMSTOT)
      ELSE
         FVRMS= 1.
      END IF

      IF (IMODE.EQ.1) THEN
         WRITE(6,3941)FVRMS,ABS(FVRMS-FVRMS0)
         WRITE(6,1021)
         WRITE(6,1022)(VEL(PNODE(N),1),N= 1,36)
         WRITE(6,1023)
         WRITE(6,1022)(VEL(PNODE(N),2),N= 1,36)
      END IF
3941 FORMAT(' FVRMS= ',E12.6,'  ABS(FVRMS-FVRMS0)= ',E12.6)

      IF (ICONV.EQ.0) THEN
         IF (FVRMS.LE.VTOL.OR.ABS(FVRMS-FVRMS0).LE.VTOL/10.) GOTO 201
      ELSE
         IF (FVRMS.LE.VTOL) GOTO 201
      END IF

      FVRMS0= FVRMS

300 CONTINUE

      IF (IOUT8.NE.0) WRITE(8,1033)
      IF (IMODE.EQ.1) WRITE(6,1033)
      STOP

C************************************************************************
C PRESSURE CORRECTION- Initial estimate
C************************************************************************
```

170

```
201  CONTINUE
     IFREEZE= 0

     DO 202 I= 1,NCORNER
     BVEC(I)= 0.
     XVEC(I)= 0.
     DP(I)= 0.
     DO 202 J= 1,NCORNER
       AMAT(I,J)= 0.
202  CONTINUE

     DO 210 NE= 1,NELEMENT

     IF (ECL(NE,1).EQ.0) THEN
     NLN= 4
     ELSE IF (ECL(NE,1).EQ.1) THEN
     NLN= NLNS
     ELSE IF (ECL(NE,1).EQ.2) THEN
     NLN= NLNC
     END IF

     DO 210 N= 1,NLN

     IF (ECL(NE,1).EQ.0) THEN
       NG= CNODE(CL(ECL(NE,2),N))
     ELSE IF (ECL(NE,1).EQ.1) THEN
       NG= CNODE(CLLQ(ECL(NE,2),N))
     ELSE IF (ECL(NE,1).EQ.2) THEN
       NG= CNODE(CLQ(ECL(NE,2),N))
     END IF

     P2N= 0.
     DO 211 M= 1,NLNQ
     DO 211 I= 1,2
       IF (ECL(NE,1).EQ.0) THEN
         P2N= P2N+EF(ECL(NE,2),N,M,I)*VEL(CQ(NE,M),I)
       ELSE IF (ECL(NE,1).EQ.1) THEN
         P2N= P2N+EFLQ(ECL(NE,2),N,M,I)*VEL(CQ(NE,M),I)
       ELSE IF (ECL(NE,1).EQ.2) THEN
         P2N= P2N+EFQ(ECL(NE,2),N,M,I)*VEL(CQ(NE,M),I)
       END IF
211    CONTINUE
     BVEC(NG)= BVEC(NG)-P2N/DELT

     DO 210 M= 1,NLN

       IF (ECL(NE,1).EQ.0) THEN
         MG= CNODE(CL(ECL(NE,2),M))
         AMAT(NG,MG)= AMAT(NG,MG)+EE(ECL(NE,2),N,M)
       ELSE IF (ECL(NE,1).EQ.1) THEN
         MG= CNODE(CLLQ(ECL(NE,2),M))
         AMAT(NG,MG)= AMAT(NG,MG)+EELQ(ECL(NE,2),N,M)
       ELSE IF (ECL(NE,1).EQ.2) THEN
         MG= CNODE(CLQ(ECL(NE,2),M))
         AMAT(NG,MG)= AMAT(NG,MG)+EEQ(ECL(NE,2),N,M)
       END IF
```

```
      210 CONTINUE

C DIRICHLET BOUNDARY CONDITIONS

          DO 220 KK= 1,2
            IF (KK.EQ.1) N= 1
            IF (KK.EQ.2) N= CNODE(NXP)
          BVEC(N)= 0.
          DO 220 K= 1,NCORNER
            IF (K.EQ.N) THEN
              AMAT(N,K)= 1.
            ELSE
              AMAT(N,K)= 0.
            END IF
      220 CONTINUE

C MATRIX SOLUTION VIA GAUSS SUBSTITUTION

          CALL GAUSS(NCORNER)

          DO 222 N= 1,NCORNER
          DP(N)= XVEC(N)
      222 CONTINUE

          IF (IMODE.EQ.1) THEN
            WRITE(6,5924)FPRMS,FPRMS
            WRITE(6,1024)
            WRITE(6,1022)(DP(CNODE(PNODE(N))),N=1,36)
          END IF

C*********************************************************************
C VELOCITY CORRECTION BASED ON PRESSURE CORRECTION
C Establishes new initial velocity values for next velocity iteration.
C*********************************************************************

      207 CONTINUE

          DO 231 ITV= 1,500

          DO 232 NG= 1,NGNODE
          DO 232 I= 1,2
      232 PREV(NG,I)= DVEL(NG,I)

          DO 230 I= 1,2
          DO 230 NG= 1,NGNODE

          IF (IDIRICH(NG).EQ.1) GOTO 230
          AMAT0= 0.
          BVEC0= 0.

          DO 240 NNE= 1,NCE(NG)

              NE= CE(NG,NNE)
              DO 244 N= 1,NLNQ
      244     IF (CQ(NE,N).EQ.NG) GOTO 245
              STOP
      245     CONTINUE
```

172

```
          SM= 0.
          IF (ECL(NE,1).EQ.0) THEN
            DO 241 M= 1,4
241           SM= SM+ED(ECL(NE,2),N,M,I)*DP(CNODE(CL(ECL(NE,2),M)))
          ELSE IF (ECL(NE,1).EQ.1) THEN
            DO 242 M= 1,NLNS
242           SM= SM+EDLQ(ECL(NE,2),N,M,I)*DP(CNODE(CLLQ(ECL(NE,2),M)))
          ELSE IF (ECL(NE,1).EQ.2) THEN
            DO 243 M= 1,NLNC
243           SM= SM+EDQ(ECL(NE,2),N,M,I)*DP(CNODE(CLQ(ECL(NE,2),M)))
          END IF
          BVEC0= BVEC0- DELT*SM

          DO 240 M= 1,NLNQ
            MG= CQ(NE,M)
            IF (MG.EQ.NG) THEN
              AMAT0= AMAT0+ EA(NE,N,M)
            ELSE
              BVEC0= BVEC0- EA(NE,N,M)*PREV(MG,I)
            END IF
240 CONTINUE

          DVEL(NG,I)= .5*(BVEC0/AMAT0-PREV(NG,I))+PREV(NG,I)

230 CONTINUE

          VRMSTOT= 0.
          VRMSERR= 0.
          VRMSERR2= 0.
          DO 233 NG= 1,NGNODE
          DO 233 I= 1,2
            VRMSTOT= VRMSTOT+VEL(NG,I)*VEL(NG,I)
            VRMSERR= VRMSERR+DVEL(NG,I)*DVEL(NG,I)
            VRMSERR2=VRMSERR2+PREV(NG,I)*PREV(NG,I)
233 CONTINUE

          IF (VRMSERR2.NE.0.) THEN
            FVRMS=ABS((SQRT(VRMSERR)-SQRT(VRMSERR2))/SQRT(VRMSERR2))
          ELSE
          FVRMS= 1.
          END IF

          IF (IMODE.EQ.1) THEN
            WRITE(6,3941)FVRMS,FVRMS
            WRITE(6,1021)
            WRITE(6,1022)(DVEL(PNODE(N),1),N= 1,36)
            WRITE(6,1023)
            WRITE(6,1022)(DVEL(PNODE(N),2),N= 1,36)
          END IF

          IF (FVRMS.LE.VTOL) GOTO 234

231 CONTINUE

C**********************************************************************
C PRESSURE CORRECTION-Last part of SIMPLER ALGORITHM
C**********************************************************************
```

173

```
234 CONTINUE

    DO 270 I= 1,NCORNER
    BVEC(I)= 0.
    XVEC(I)= 0.
    DP(I)= 0.
    DO 270 J= 1,NCORNER
      AMAT(I,J)= 0.
270 CONTINUE

    DO 280 NE= 1,NELEMENT

    IF (ECL(NE,1).EQ.0) THEN
    NLN= 4
    ELSE IF (ECL(NE,1).EQ.1) THEN
    NLN= NLNS
    ELSE IF (ECL(NE,1).EQ.2) THEN
    NLN= NLNC
    END IF

    DO 280 N= 1,NLN

    IF (ECL(NE,1).EQ.0) THEN
    NG= CNODE(CL(ECL(NE,2),N))
    ELSE IF (ECL(NE,1).EQ.1) THEN
    NG= CNODE(CLLQ(ECL(NE,2),N))
    ELSE IF (ECL(NE,1).EQ.2) THEN
    NG= CNODE(CLQ(ECL(NE,2),N))
    END IF

    SMI= 0.
    DO 281 M= 1,NLNQ
    DO 281 I= 1,2
    IF (ECL(NE,1).EQ.0) THEN
       SMI= SMI+ EF(ECL(NE,2),N,M,I)*VEL(CQ(NE,M),I)
    ELSE IF (ECL(NE,1).EQ.1) THEN
       SMI= SMI+ EFLQ(ECL(NE,2),N,M,I)*VEL(CQ(NE,M),I)
    ELSE IF (ECL(NE,1).EQ.2) THEN
       SMI= SMI+ EFQ(ECL(NE,2),N,M,I)*VEL(CQ(NE,M),I)
    END IF
281 CONTINUE
    BVEC(NG)= BVEC(NG)- SMI/DELT

    SUM= 0.
    DO 282 GAM= 1,NLNQ
    DO 282 J= 1,2
    SMI= 0.
    DO 283 M= 1,NLNQ
    DO 283 I= 1,2
      IF (ECL(NE,1).EQ.0) THEN
        SMI= SMI+ EG(ECL(NE,2),N,M,J,GAM,I)*DVEL(CQ(NE,M),I)
      ELSE IF (ECL(NE,1).EQ.1) THEN
        SMI= SMI+ EGLQ(ECL(NE,2),N,M,J,GAM,I)*DVEL(CQ(NE,M),I)
      ELSE IF (ECL(NE,1).EQ.2) THEN
        SMI= SMI+ EGQ(ECL(NE,2),N,M,J,GAM,I)*DVEL(CQ(NE,M),I)
      END IF
283    CONTINUE
282 SUM= SUM+ SMI*(2.*VEL(CQ(NE,GAM),J)+DVEL(CQ(NE,GAM),J))
```

```fortran
      BVEC(NG)= BVEC(NG)+ SUM

      SUM= 0.
      DO 284 M= 1,NLNQ
      DO 284 J= 1,2
      SGI= 0.
      DO 285 GAM= 1,NLNQ
      DO 285 I= 1,2
        IF (ECL(NE,1).EQ.0) THEN
          SGI= SGI+ EH(ECL(NE,2),N,M,GAM,I,J)*DVEL(CQ(NE,GAM),I)
        ELSE IF (ECL(NE,1).EQ.1) THEN
          SGI= SGI+ EHLQ(ECL(NE,2),N,M,GAM,I,J)*DVEL(CQ(NE,GAM),I)
        ELSE IF (ECL(NE,1).EQ.2) THEN
          SGI= SGI+ EHQ(ECL(NE,2),N,M,GAM,I,J)*DVEL(CQ(NE,GAM),I)
        END IF
  285   CONTINUE
  284 SUM= SUM+ SGI*VEL(CQ(NE,M),J)
      BVEC(NG)= BVEC(NG)+ SUM

      DO 280 M= 1,NLN

      IF (ECL(NE,1).EQ.0) THEN
        MG= CNODE(CL(ECL(NE,2),M))
        AMAT(NG,MG)= AMAT(NG,MG)+EE(ECL(NE,2),N,M)
      ELSE IF (ECL(NE,1).EQ.1) THEN
        MG= CNODE(CLLQ(ECL(NE,2),M))
        AMAT(NG,MG)= AMAT(NG,MG)+EELQ(ECL(NE,2),N,M)
      ELSE IF (ECL(NE,1).EQ.2) THEN
        MG= CNODE(CLQ(ECL(NE,2),M))
        AMAT(NG,MG)= AMAT(NG,MG)+EEQ(ECL(NE,2),N,M)
      END IF

  280 CONTINUE

C Pressure Correction Dirichlet Condition

      DO 290 KK= 1,2
        IF (KK.EQ.1) N= 1
        IF (KK.EQ.2) N= CNODE(NXP)
      BVEC(N)= 0.
      DO 290 K= 1,NCORNER
        IF (K.EQ.N) THEN
          AMAT(N,K)= 1.
        ELSE
          AMAT(N,K)= 0.
        END IF
  290 CONTINUE

C MATRIX SOLUTION VIA GAUSS SUBSTITUTION

      CALL GAUSS(NCORNER)

C Error and convergence tolerance calculation

      PRMSERR= 0.
      PRMSTOT= 0.
      DO 292 N= 1,NCORNER
        PRMSTOT= PRMSTOT+(PDYN(GNODE(N))-PDYN(NGNODE))*
```

```
      >                 (PDYN(GNODE(N))-PDYN(NGNODE))
          PRMSERR= PRMSERR+XVEC(N)*XVEC(N)
          PDYN(GNODE(N))= PDYN(GNODE(N))+XVEC(N)
      292 CONTINUE

          VRMSERR= 0.
          VRMSTOT= 0.
          DO 293 N= 1,NGNODE
          DO 293 I= 1,2
             VRMSERR= VRMSERR+DVEL(N,I)*DVEL(N,I)
             VRMSTOT= VRMSTOT+VEL(N,I)*VEL(N,I)
             VEL(N,I)= DVEL(N,I)+VEL(N,I)
      293 CONTINUE

          IF (VRMSTOT.NE.0.) THEN
             FVRMS2= SQRT(VRMSERR/VRMSTOT)
          ELSE
             FVRMS2= 1.
          END IF

          IF (PRMSTOT.NE.0.) THEN
             FPRMS= SQRT(PRMSERR/PRMSTOT)
          ELSE
             FPRMS= 1.
          END IF

          IF (IMODE.EQ.1) WRITE(6,5924)FPRMS,ABS(FPRMS-FPRMS0)
      5924 FORMAT(' FPRMS= ',E12.6,' ABS(FPRMS-FPRMS0)=    ',E12.6)
          IF (IMODE.EQ.1) WRITE(6,5925)FVRMS2,ABS(FVRMS2-FVRMS20)
      5925 FORMAT(' FVRMS2= ',E12.6,' ABS(FVRMS2-FVRMS20)= ',E12.6)

          IF (FPRMS.LE.PTOL.OR.ABS(FPRMS-FPRMS0).LE.PTOL) GOTO 101
          IF (FVRMS2.LE.VTOL) GOTO 101
          FPRMS0= FPRMS
          FVRMS20= FVRMS2

          IF (IMODE.EQ.1) THEN
             WRITE(6,1024)
             WRITE(6,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
          END IF

      200 CONTINUE

          IF (IMODE.EQ.1) WRITE(6,1034)
          IF (IOUT8.NE.0) WRITE(8,1034)
          STOP

C*********************************************************************
C TEMPERATURE SOLUTION
C*********************************************************************

      101 CONTINUE

          DO 104 NG= 1,NGNODE2
          PREV(NG,1)= TEMP(NG)
      104 PREV(NG,2)= TEMP(NG)

          DO 105 ITT= 1,1000
```

176

```
         DO 106 NG= 1,NGNODE2
         IF (IDIRICH(NG).EQ.1) GOTO 106
         AMAT0= 0.
         BVEC0= 0.

         DO 110 NNE= 1,NCE(NG)

         NE= CE(NG,NNE)
         IF (ECL(NE,1).EQ.0) THEN
           NM= NLNQ
         ELSE IF (ECL(NE,1).EQ.1) THEN
           NM= 15
         ELSE IF (ECL(NE,1).EQ.2) THEN
           NM= 25
         END IF
         DO 108 N= 1,NM
            IF (ECL(NE,1).EQ.0) THEN
               IF (CQ(NE,N).EQ.NG) GOTO 109
            ELSE IF (ECL(NE,1).EQ.1) THEN
               IF (C24(ECL(NE,2),N).EQ.NG) GOTO 109
            ELSE IF (ECL(NE,1).EQ.2) THEN
               IF (C44(ECL(NE,2),N).EQ.NG) GOTO 109
            END IF
108      CONTINUE
         STOP
109      CONTINUE

         IF (ITIME.EQ.1.AND.ITTEMP.EQ.1.AND.ISSTEMP.EQ.1) THEN
            IF (ITMEN.EQ.1) THEN
               GOTO 114
            ELSE
               GOTO 9998
            END IF
         END IF

         TRESID= 0.
         DO 111 M= 1,NM
         MNM1= 0.
         IF (NM.EQ.9) THEN
            DO 112 GAM= 1,NLNQ
            DO 112 J= 1,2
112         MNM1=MNM1+EB(NE,N,GAM,M,J)*VEL0(CQ(NE,GAM),J)
            TRESID= TRESID+TEMP0(CQ(NE,M))*
     *         (EA(NE,N,M)-DELT*FT0*(MNM1+EC(NE,N,M)/PR))
         ELSE IF (NM.EQ.15) THEN
            DO 115 GAM= 1,NLNQ
            DO 115 J= 1,2
115         MNM1=MNM1+EB24(ECL(NE,2),N,GAM,M,J)*VEL0(CQ(NE,GAM),J)
            TRESID= TRESID+TEMP0(C24(ECL(NE,2),M))*
     *         (EA24T(ECL(NE,2),N,M)-
     *         DELT*FT0*(MNM1+EC24(ECL(NE,2),N,M)/PR))
         ELSE
            DO 116 GAM= 1,NLNQ
            DO 116 J= 1,2
116         MNM1=MNM1+EB44(ECL(NE,2),N,GAM,M,J)*VEL0(CQ(NE,GAM),J)
            TRESID= TRESID+TEMP0(C44(ECL(NE,2),M))*
     *         (EA44T(ECL(NE,2),N,M)-
```

```
*      DELT*FT0*(MNM1+EC44(ECL(NE,2),N,M)/PR))
       END IF
111 CONTINUE

    BVEC0= BVEC0+TRESID

114 CONTINUE

    DO 110 M= 1,NM

    MNM1= 0.
    IF (NM.EQ.9) THEN
    MG= CQ(NE,M)
    DO 113 GAM= 1,NLNQ
    DO 113 J= 1,2
113    MNM1=MNM1+EB(NE,N,GAM,M,J)*VEL(CQ(NE,GAM),J)
    ELSE IF (NM.EQ.15) THEN
    MG= C24(ECL(NE,2),M)
    DO 117 GAM= 1,NLNQ
    DO 117 J= 1,2
117    MNM1=MNM1+EB24(ECL(NE,2),N,GAM,M,J)*VEL(CQ(NE,GAM),J)
    ELSE
    MG= C44(ECL(NE,2),M)
    DO 118 GAM= 1,NLNQ
    DO 118 J= 1,2
118    MNM1=MNM1+EB44(ECL(NE,2),N,GAM,M,J)*VEL(CQ(NE,GAM),J)
    END IF

    IF (ITIME.EQ.1.AND.ITTEMP.EQ.1.AND.ISSTEMP.EQ.1) THEN
    IF (NG.EQ.MG) THEN
      IF (NM.EQ.9) THEN
        AMAT0= AMAT0+EC(NE,N,M)/PR
      ELSE IF (NM.EQ.15) THEN
        AMAT0= AMAT0+EC24(ECL(NE,2),N,M)/PR
      ELSE
        AMAT0= AMAT0+EC44(ECL(NE,2),N,M)/PR
      END IF
    ELSE
      IF (NM.EQ.9) THEN
        BVEC0= BVEC0-EC(NE,N,M)/PR*PREV(MG,1)
      ELSE IF (NM.EQ.15) THEN
        BVEC0= BVEC0-EC24(ECL(NE,2),N,M)/PR*PREV(MG,1)
      ELSE
        BVEC0= BVEC0-EC44(ECL(NE,2),N,M)/PR*PREV(MG,1)
      END IF
    END IF
    ELSE
    IF (NG.EQ.MG) THEN
      IF (NM.EQ.9) THEN
        AMAT0= AMAT0+EA(NE,N,M)+
*          FT1*DELT*(MNM1+EC(NE,N,M)/PR)
      ELSE IF (NM.EQ.15) THEN
        AMAT0= AMAT0+EA24T(ECL(NE,2),N,M)+
*          FT1*DELT*(MNM1+EC24(ECL(NE,2),N,M)/PR)
      ELSE
        AMAT0= AMAT0+EA44T(ECL(NE,2),N,M)+
*          FT1*DELT*(MNM1+EC44(ECL(NE,2),N,M)/PR)
      END IF
```

```
                ELSE
                   IF (NM.EQ.9) THEN
                      BVEC0= BVEC0-PREV(MG,1)*(EA(NE,N,M)+
             *          FT1*DELT*(MNM1+EC(NE,N,M)/PR))
                   ELSE IF (NM.EQ.15) THEN
                      BVEC0= BVEC0-PREV(MG,1)*(EA24T(ECL(NE,2),N,M)+
             *          FT1*DELT*(MNM1+EC24(ECL(NE,2),N,M)/PR))
                   ELSE
                      BVEC0= BVEC0-PREV(MG,1)*(EA44T(ECL(NE,2),N,M)+
             *          FT1*DELT*(MNM1+EC44(ECL(NE,2),N,M)/PR))
                   END IF
                END IF
                END IF

         110 CONTINUE

    C TEMPERATURE BOUNDARY CONDITIONS

    C MIXED TEMPERATURE BC FOR SURFACE 4

             IF ((NG.LE.NXP).OR.(NG.GE.NGNODE+1.AND.NG.LE.NGNODE+4)) THEN
             IF (NG.LE.NXP) THEN
                NP= NCES(NG)
             ELSE
                NP= 1
             END IF
             DO 120 NNE= 1,NP
                IF (NG.LE.NXP) THEN
                   NE= CES(NG,NNE)
                ELSE IF (NG.LE.NGNODE+2) THEN
                   NE= 1
                ELSE
                   NE= NESURF(4)
                END IF
                IF (NE.EQ.1.OR.NE.EQ.NESURF(4)) THEN
                   DO 124 N= 1,5
         124          IF (C44(ECL(NE,2),14-N).EQ.NG) GOTO 125
                   STOP
         125       CONTINUE
                   IF (ITIME.EQ.1.AND.ITTEMP.EQ.1.AND.ISSTEMP.EQ.1) THEN
                    DCOEFF= -1.*BI/PR*SURFL(4,NE)/5670.
                      DCOEFF2= DCOEFF
                    IF (ITYPE.EQ.1)
             >           BVEC0=BVEC0+BI/PR*SURFL(4,NE)*SC2(N)/90.
                   ELSE
                    DCOEFF= -1.*BI/PR*SURFL(4,NE)/5670.*DELT*FT1
                    DCOEFF2= -1.*BI/PR*SURFL(4,NE)/5670.*DELT*FT0
                    IF (ITYPE.EQ.1)
             >           BVEC0=BVEC0+BI/PR*SURFL(4,NE)*SC2(N)/90.*DELT
                   END IF
                   CON= 0.
                   DO 126 M= 1,5
                      MG= C44(ECL(NE,2),14-M)
                      IF (MG.EQ.1.OR.MG.EQ.NXP) GOTO 126
                      CON= CON+DCOEFF2*SB2(N,M)*TEMP0(MG)
                      IF (MG.EQ.NG) THEN
                         AMAT0= AMAT0-DCOEFF*SB2(N,M)
                      ELSE
```

```fortran
             BVEC0= BVEC0+DCOEFF*SB2(N,M)*PREV(MG,1)
             END IF
126          CONTINUE
       ELSE
         DO 121 N= 1,3
121        IF (CSURF(4,NE,N).EQ.NG) GOTO 122
         STOP
122        CONTINUE
         IF (ITIME.EQ.1.AND.ITTEMP.EQ.1.AND.ISSTEMP.EQ.1) THEN
          DCOEFF= -1.*BI/PR*SURFL(4,NE)/30
            DCOEFF2= DCOEFF
          IF (ITYPE.EQ.1)BVEC0=BVEC0+BI/PR*SURFL(4,NE)*SC(N)/6.
         ELSE
          DCOEFF= -1.*BI/PR*SURFL(4,NE)/30.*DELT*FT1
          DCOEFF2= -1.*BI/PR*SURFL(4,NE)/30.*DELT*FT0
          IF (ITYPE.EQ.1)BVEC0=BVEC0+BI/PR*SURFL(4,NE)*SC(N)/6.*DELT
         END IF
         CON= 0.
         DO 123 M= 1,3
           MG= CSURF(4,NE,M)
           IF (MG.EQ.1.OR.MG.EQ.NXP) GOTO 123
           CON= CON+DCOEFF2*SB(N,M)*TEMP0(MG)
           IF (MG.EQ.NG) THEN
             AMAT0= AMAT0-DCOEFF*SB(N,M)
           ELSE
             BVEC0= BVEC0+DCOEFF*SB(N,M)*PREV(MG,1)
           END IF
123        CONTINUE
       END IF
     IF (ITIME.NE.1.OR.ITTEMP.NE.1.OR.ISSTEMP.NE.1)
   >   BVEC0= BVEC0+CON
120 CONTINUE
     END IF

     TEMP(NG)= .3*(BVEC0/AMAT0-PREV(NG,1))+PREV(NG,1)

106 CONTINUE

     TOTV= 0.
     VERR= 0.
     DO 9215 NG= 1,NGNODE2
     TOTV= TOTV+PREV(NG,1)*PREV(NG,1)
     ERR= TEMP(NG)-PREV(NG,1)
     VERR= VERR+ERR*ERR
     PREV(NG,1)= TEMP(NG)
9215 CONTINUE

     IF (TOTV.NE.0.) THEN
       FERR= SQRT(VERR/TOTV)
     ELSE
       FERR= 1.
     END IF

     IF (IMODE.EQ.1) THEN
     WRITE(6,1080) FERR
     WRITE(6,1022)(TEMP(PNODE(N)),N= 1,36)
     END IF
1080 FORMAT(' TEMP SOLUTION LOOP/ ERROR= ',E12.6)
```

C-3

```
            IF (FERR.LE.TTOL) GOTO 119
      105 CONTINUE
      119 CONTINUE

            TOTV= 0.
            VERR= 0.
            ERRAVG= 0.
            ERRMIN= ABS(TEMP(2)-PREV(2,2))
            ERRMAX= ERRMIN
            DO 150 NG= 1,NGNODE2
            TOTV= TOTV+PREV(NG,2)*PREV(NG,2)
            ERR= TEMP(NG)-PREV(NG,2)
              IF (ERR.NE.0.) THEN
                IF (ABS(ERR).LT.ERRMIN)ERRMIN= ABS(ERR)
                IF (ABS(ERR).GT.ERRMAX)ERRMAX= ABS(ERR)
              END IF
              ERRAVG= ERRAVG+ ABS(ERR)
            VERR= VERR+ERR*ERR
      150 CONTINUE
            ERRAVG= ERRAVG/REAL(NGNODE2)

            IF (TOTV.NE.0.) THEN
              FERR= SQRT(VERR/TOTV)
            ELSE
              FERR= 1.
            END IF

            IF (IMODE.EQ.1) THEN
            WRITE(6,1081) FERR
            WRITE(6,1025)
            WRITE(6,1022)(TEMP(PNODE(N)),N= 1,36)
            END IF
     1081 FORMAT(' TEMP/TIME STEP ERROR= ',E12.6)

            IF (ITIME.EQ.1.AND.ITTEMP.EQ.1) THEN
            IF (FERR.GT.TTOL) THEN
              GOTO 101
            ELSE
              GOTO 9998
            END IF
            END IF

            IF (FERR.LE.TTOL) GOTO 901

      100 CONTINUE

            IF (IOUT8.NE.0) WRITE(8,1043)
            IF (IMODE.EQ.1) WRITE(6,1043)
            STOP

C*********************************************************************
C TIME STEP OUTPUT AND INITIALIZATION
C*********************************************************************

      901 CONTINUE

            TIME= TIME+DELT
```

```
          IF (IOUT8.GE.2) THEN
            WRITE(8,1030)ITIME,TIME
            WRITE(8,1021)
            WRITE(8,1022)(VEL(PNODE(N),1),N= 1,36)
            WRITE(8,1023)
            WRITE(8,1022)(VEL(PNODE(N),2),N= 1,36)
            WRITE(8,1024)
            WRITE(8,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
            WRITE(8,1025)
            WRITE(8,1022)(TEMP(PNODE(N)),N= 1,36)
          END IF
          IF (IMODE.EQ.1) THEN
            WRITE(6,1030)ITIME,TIME
            WRITE(6,1021)
            WRITE(6,1022)(VEL(PNODE(N),1),N= 1,36)
            WRITE(6,1023)
            WRITE(6,1022)(VEL(PNODE(N),2),N= 1,36)
            WRITE(6,1024)
            WRITE(6,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
            WRITE(6,1025)
            WRITE(6,1022)(TEMP(PNODE(N)),N= 1,36)
          END IF

C Check steady-state convergence and initialize old values for
C next time step

 9998 CONTINUE

          TSSCON= 0.
          PSSCON= 0.
          VSSCON= 0.
          TSUM= 0.
          PSUM= 0.
          VSUM= 0.
          DO 910 I= 1,NGNODE
          PSUM= PSUM+(PDYN0(I)-PDYN0(NGNODE))*(PDYN0(I)-PDYN0(NGNODE))
          PSSCON= PSSCON+(PDYN(I)-PDYN0(I))*(PDYN(I)-PDYN0(I))
          PDYN0(I)= PDYN(I)
          DO 910 J= 1,2
            VSUM= VSUM+VEL0(I,J)*VEL0(I,J)
              ERR= VEL(I,J)-VEL0(I,J)
            VSSCON= VSSCON+ERR*ERR
            VEL0(I,J)= VEL(I,J)
  910 CONTINUE
          DO 911 I= 1,NGNODE2
          TSUM= TSUM+TEMP0(I)*TEMP0(I)
          TSSCON= TSSCON+(TEMP(I)-TEMP0(I))*(TEMP(I)-TEMP0(I))
          TEMP0(I)= TEMP(I)
  911 CONTINUE
          IF (TSUM.NE.0.) THEN
          TSSCON= SQRT(TSSCON/TSUM)
          ELSE
          TSSCON= 1.
          END IF
          IF (PSUM.NE.0.) THEN
          PSSCON= SQRT(PSSCON/PSUM)
          ELSE
          PSSCON= 1.
```

182

```fortran
      END IF
      IF (VSUM.NE.0.) THEN
      VSSCON= SQRT(VSSCON/VSUM)
      ELSE
      VSSCON= 1.
      END IF

      IF (IOUT8.GE.2) WRITE(8,1036) VSSCON,PSSCON,TSSCON
      IF (IMODE.EQ.1) WRITE(6,1036) VSSCON,PSSCON,TSSCON

      IF (MA.GE.MAIN.AND.RINV.GE.1./REVAPIN) THEN
        IF (TSSCON.LE.SSTOL.AND.ITIME.GE.5) ICONV= 1
        IF (ITMEN.EQ.1) THEN
          IF (TSSCON.LE.SSTOL.AND.ITIME.GE.200) ITCONV= 1
        ELSE
          IF (TSSCON.LE.SSTOL.AND.ITIME.GE.30) ITCONV= 1
        END IF
      END IF

C Intermittent update and continuation check

      IF (IMODE.EQ.2) THEN

      IF (QUETYPE.EQ.'HUGE3') THEN
        OPEN(UNIT=2,FILE='huge3.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='huge3.dat',STATUS='UNKNOWN')
          OPEN(UNIT=1,FILE='huge3.ts2',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'HUGE2') THEN
        OPEN(UNIT=2,FILE='huge2.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='huge2.dat',STATUS='UNKNOWN')
          OPEN(UNIT=1,FILE='huge2.ts2',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE4') THEN
        OPEN(UNIT=2,FILE='large4.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='large4.dat',STATUS='UNKNOWN')
          OPEN(UNIT=1,FILE='large4.ts2',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE3') THEN
        OPEN(UNIT=2,FILE='large3.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='large3.dat',STATUS='UNKNOWN')
          OPEN(UNIT=1,FILE='large3.ts2',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE2') THEN
        OPEN(UNIT=2,FILE='large2.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='large2.dat',STATUS='UNKNOWN')
          OPEN(UNIT=1,FILE='large2.ts2',STATUS='UNKNOWN')
      ELSE
        OPEN(UNIT=2,FILE='other.tst',STATUS='OLD')
        OPEN(UNIT=3,FILE='other.dat',STATUS='UNKNOWN')
          OPEN(UNIT=1,FILE='other.ts2',STATUS='UNKNOWN')
      END IF

      READ(2,1003) TITLINE,IGO
      CLOSE(2)

        WRITE(3,3919)ITMEN,YERR
        WRITE(3,1030)ITIME,TIME
        WRITE(3,6391)VSSCON,PSSCON,TSSCON
          WRITE(3,7391)ERRAVG
          WRITE(3,7392)ERRMIN
          WRITE(3,7393)ERRMAX
```

```fortran
7391 FORMAT(' ERRAVG= ',E12.6)
7392 FORMAT(' ERRMIN= ',E12.6)
7393 FORMAT(' ERRMAX= ',E12.6)
      WRITE(3,1021)
      WRITE(3,1022)(VEL(PNODE(N),1),N= 1,36)
      WRITE(3,1023)
      WRITE(3,1022)(VEL(PNODE(N),2),N= 1,36)
      WRITE(3,1024)
      WRITE(3,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
      WRITE(3,1025)
      WRITE(3,1022)(TEMP(PNODE(N)),N= 1,36)
      CLOSE(3)

      IF (IGO.EQ.0) THEN
        IF (QUETYPE.EQ.'HUGE3') THEN
          OPEN(UNIT=2,FILE='huge3.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'HUGE2') THEN
          OPEN(UNIT=2,FILE='huge2.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'LARGE4') THEN
          OPEN(UNIT=2,FILE='large4.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'LARGE3') THEN
          OPEN(UNIT=2,FILE='large3.tst',STATUS='UNKNOWN')
        ELSE IF (QUETYPE.EQ.'LARGE2') THEN
          OPEN(UNIT=2,FILE='large2.tst',STATUS='UNKNOWN')
        ELSE
          OPEN(UNIT=2,FILE='other.tst',STATUS='UNKNOWN')
        END IF
        WRITE(2,1003) TITLINE,1
        CLOSE(2)
          IF (IOUT9.EQ.1) THEN
            TITLINE=' MENISCUS DIM CHARACTERISTICS'
            WRITE(9,1073) TITLINE,0
            DO 931 K= 1,NXP
              WRITE(9,1009) XSURF(K),YSTAT(K),YSURF(K),
     >        SURFNN(K,1),SURFNN(K,2),SURFNT(K,1),SURFNT(K,2)
931         CONTINUE
          END IF
        GOTO 510
      END IF

      READ(1,1003) TITLINE,ICHK
      CLOSE(1)
      IF (ITCONV.EQ.1) ICHK= 1   ! Full data dump to U10 when t conv
      IF (ICHK.EQ.1) THEN
      IF (QUETYPE.EQ.'HUGE3') THEN
        OPEN(UNIT=1,FILE='huge3.ts2',STATUS='UNKNOWN')
          OPEN(UNIT=10,FILE='chg3.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'HUGE2') THEN
        OPEN(UNIT=1,FILE='huge2.ts2',STATUS='UNKNOWN')
          OPEN(UNIT=10,FILE='chg2.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE4') THEN
        OPEN(UNIT=1,FILE='large4.ts2',STATUS='UNKNOWN')
          OPEN(UNIT=10,FILE='clrg4.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE3') THEN
        OPEN(UNIT=1,FILE='large3.ts2',STATUS='UNKNOWN')
          OPEN(UNIT=10,FILE='clrg3.dat',STATUS='UNKNOWN')
      ELSE IF (QUETYPE.EQ.'LARGE2') THEN
        OPEN(UNIT=1,FILE='large2.ts2',STATUS='UNKNOWN')
```

184

```
               OPEN(UNIT=10,FILE='clrg2.dat',STATUS='UNKNOWN')
          ELSE
            OPEN(UNIT=1,FILE='other.ts2',STATUS='UNKNOWN')
              OPEN(UNIT=10,FILE='coth.dat',STATUS='UNKNOWN')
          END IF
            WRITE(1,1003)TITLINE,0
            CLOSE(1)
            WRITE(10,8294)QUETYPE
 8294 FORMAT(A12,28X,',',19X,'0')
            WRITE(10,8295)ITMEN
            WRITE(10,8296)ITIME
            WRITE(10,8297)VSSCON
            WRITE(10,8298)PSSCON
            WRITE(10,8299)TSSCON
 8295 FORMAT('Meniscus Iteration                              ,',I20)
 8296 FORMAT('Time Step                                       ,',I20)
 8297 FORMAT('Velocity Error                                  ,',1PE20.4)
 8298 FORMAT('Press Error                                     ,',1PE20.4)
 8299 FORMAT('Temperature Error/SS Conv Test                  ,',1PE20.4)
            WRITE(10,1072)
            DO 8301 I=1,3
 8301       WRITE(10,1073)TITLINE,NCONTOUR
            DO 8302 I=1,3
 8302       WRITE(10,1074)TITLINE,XARROW
            TITLINE=' Number of top nodes                  '
          WRITE(10,1073)TITLINE,2*NXE+1
            TITLINE=' Number of global nodes               '
          WRITE(10,1073)TITLINE,(2*NXE+1)*(2*NYE+1)
            TITLINE=' MENISCUS DIM CHARACTERISTICS'
          WRITE(10,1073)TITLINE,0
            DO 8341 K= 1,NXP
              WRITE(10,1009)XSURF(K),YSTAT(K),YSURF(K),SURFNN(K,1),
     >                      SURFNN(K,2),SURFNT(K,1),SURFNT(K,2)
 8341       CONTINUE
            TITLINE= ' MOST RECENT TIME INTERVAL'
            WRITE(10,1079)TITLINE,DELT
            WRITE(10,6295)
            CALL STREAM(NGNODE,NXP,NYE+1,2*NXP)
            DO 8342 NG= 1,NGNODE
              IF (NG.LE.NXP) THEN
                WRITE(10,6297)NG,XG(NG,1),XG(NG,2),VEL(NG,1),VEL(NG,2),
     >          TEMP(NG),PBUB(NG),PSI(NG),PDYN(NG)-PDYN(NGNODE)
              ELSE
                WRITE(10,6297)NG,XG(NG,1),XG(NG,2),VEL(NG,1),VEL(NG,2),
     >          TEMP(NG),PTOT,PSI(NG),PDYN(NG)-PDYN(NGNODE)
              END IF
 8342       CONTINUE
            CLOSE(10)

          END IF

        END IF

        IF (ITCONV.EQ.1) GOTO 510

  900 CONTINUE

        IF (IOUT8.GE.1) WRITE(8,1039)
```

```
      IF (IMODE.EQ.1) WRITE(6,1039)
C       STOP

C*********************************************************************
C RE-INITIALIZATION FOR MENISCUS CALCULATION
C*********************************************************************

  510 CONTINUE

      CALL STREAM(NGNODE,NXP,NYE+1,2*NXP)

 6391 FORMAT(' VSSCON= ',E12.6,/,' PSSCON= ',E12.6,/,
     >         ' TSSCON= ',E12.6)

      IF (IOUT8.GE.1) THEN
        WRITE(8,3919)ITMEN,YERR
        WRITE(8,1030)ITIME,TIME
        WRITE(8,6391)VSSCON,PSSCON,TSSCON
        WRITE(8,1021)
        WRITE(8,1022)(VEL(PNODE(N),1),N= 1,36)
        WRITE(8,1023)
        WRITE(8,1022)(VEL(PNODE(N),2),N= 1,36)
        WRITE(8,1050)
        WRITE(8,1022)(PSI(PNODE(N)),N= 1,36)
        WRITE(8,1024)
        WRITE(8,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
        WRITE(8,1025)
        WRITE(8,1022)(TEMP(PNODE(N)),N= 1,36)
      END IF
      IF (IMODE.EQ.1) THEN
        WRITE(6,3919)ITMEN,YERR
        WRITE(6,1030)ITIME,TIME
        WRITE(6,1021)
        WRITE(6,1022)(VEL(PNODE(N),1),N= 1,36)
        WRITE(6,1023)
        WRITE(6,1022)(VEL(PNODE(N),2),N= 1,36)
        WRITE(6,1050)
        WRITE(6,1022)(PSI(PNODE(N)),N= 1,36)
        WRITE(6,1024)
        WRITE(6,1022)(PDYN(PNODE(N))-PDYN(NGNODE),N= 1,36)
        WRITE(6,1025)
        WRITE(6,1022)(TEMP(PNODE(N)),N= 1,36)
      END IF
 3919 FORMAT(/' MEN IT# = ',I6,'   MEN ERR= ',E12.6)

      IF (IGO.EQ.0) GOTO 503

      N= NGNODE-NXP
      DO 520 K= 1,NXP
      N= N+1
      TSURF(K)= DBLE(TEMP(K))
      PSURF(K)= DBLE(PDYN(K)-PDYN(NGNODE))
  520 CONTINUE

  500 CONTINUE

      IF (IOUT8.GE.1) WRITE(8,7234)
      IF (IMODE.EQ.1) WRITE(6,7234)
```

186

```
 7234 FORMAT(' MENISCUS ITERATION DID NOT CONVERGE')
      STOP

  503 CONTINUE

C**********************************************************************
C**********************************************************************
C UNIT 9 PLOT DATA OUTPUT
C**********************************************************************
C**********************************************************************

      TITLINE= ' MOST RECENT TIME INTERVAL'
      WRITE(9,1079)TITLINE,DELT
      IF (IOUT9.GE.1) THEN
        WRITE(9,6295)
      DO 6296 NG= 1,NGNODE
         IF (NG.LE.NXP) THEN
            WRITE(9,6297)NG,XG(NG,1),XG(NG,2),VEL(NG,1),VEL(NG,2),
     >      TEMP(NG),PBUB(NG),PSI(NG),PDYN(NG)-PDYN(NGNODE)
         ELSE
            WRITE(9,6297)NG,XG(NG,1),XG(NG,2),VEL(NG,1),VEL(NG,2),
     >      TEMP(NG),PTOT,PSI(NG),PDYN(NG)-PDYN(NGNODE)
         END IF
 6296    CONTINUE
      END IF

C**********************************************************************
C**********************************************************************
C FORMAT STATEMENTS
C**********************************************************************
C**********************************************************************

 1000 FORMAT(' Type in the name of your input file: ',$)
 1001 FORMAT(' UNIT 7 INPUT FILE: ',A12,' UNIT 4 INPUT: ',A12)
 1002 FORMAT(//,3(A80,/))
 1003 FORMAT(A40,I20)
 1004 FORMAT(A40,8X,A12)
 1005 FORMAT(A40,1PE20.4)
 1020 FORMAT(' INITIAL VELOCITIES,PRESSURES AND TEMPERATURES')
 1021 FORMAT(' X1-COMPONENT VELOCITY')
 1022 FORMAT(6(1X,6(E11.5,1X),/),/)
 1023 FORMAT(' X2-COMPONENT VELOCITY')
 1024 FORMAT(' DYNAMIC PRESSURE')
 1050 FORMAT(' STREAM FUNCTION')
 1051 FORMAT(' TOTAL PRESSURE')
 1025 FORMAT(' TEMPERATURE')
 1030 FORMAT(/' ITIME=   ',I12,'   TIME= ',E12.6)
 1031 FORMAT(' PRESS CORR ITPRESS= ',I12)
 1098 FORMAT(' PRESS ITPRESS2= ',I12)
 1032 FORMAT(' ITVEL=   ',I12)
 1033 FORMAT(' VELOCITY ITERATION LOOP DID NOT CONVERGE')
 1034 FORMAT(' PRESSURE ITERATION LOOP DID NOT CONVERGE')
 1035 FORMAT(' ITTEMP=  ',I12)
 1036 FORMAT(' VEL STEADY-STATE CONV ERROR (VSSCON)=   ',E12.6,/,
     >        ' PRESS STEADY-STATE CONV ERROR (PSSCON)= ',E12.6,/,
     >        ' TEMP STEADY-STATE CONV ERROR (TSSCON)=  ',E12.6)
 1037 FORMAT(' PRESSURE ERROR=    ',E12.6)
 1038 FORMAT(/,' TIME STEP NUMBER= ',I4)
```

```
1039 FORMAT(' STEADY-STATE ITERATION FAILED TO CONVERGE')
1040 FORMAT(A12)
1041 FORMAT(/' STREAM FUNCTION')
1042 FORMAT(' ITTEMP= ',I12)
1043 FORMAT(' TEMPERATURE ITERATION LOOP DID NOT CONVERGE')
1070 FORMAT(A12/A12)
1071 FORMAT(A40,6I5)
1072 FORMAT(' PLOT DATA, 0')
1073 FORMAT(A40,',',I20)
1074 FORMAT(A40,',',1PE20.4)
1075 FORMAT(A40,1X,I20)
6220 FORMAT(A40)
1009 FORMAT(E12.6,6(',',E12.6))
1076 FORMAT(7(E12.6,1X))
1007 FORMAT(/,' MENISCUS ITERATION= ',I3,/,' MENISCUS GEOMETRY',/,
    >        2X,'I',7X,'X',7X,'Y',6X,'n1',6X,'n2',6X,'s1',6X,'s2')
1008 FORMAT(I3,6F8.4)
6295 FORMAT(' ND,        X1,        X2,        VEL1,       VEL2,    ',
    > '  TEMP,       PBUB,      PSI,        PDYN')
6297 FORMAT(I5,',',7(E10.4,',',),E10.4)
1077 FORMAT(I5,1X,7(E10.4,1X),E10.4)
1078 FORMAT(A40,1X,E20.6)
1079 FORMAT(A40,',',E20.6)

     IF (IMODE.GT.0) CLOSE(7)
     IF (IOUT8.NE.0.AND.IMODE.GT.0) CLOSE(8)
     IF (IOUT9.NE.0.AND.IMODE.GT.0) CLOSE(9)

     STOP
     END
```

B.   Subroutine STREAM

```
     SUBROUTINE STREAM(NGNODE,NXP,NYP,NROW)

     PARAMETER (MXE= 25,        MYE= 20) ! # elements in x & y direcs
     PARAMETER (MXN= 1+2*MXE)            ! # nodes in x-direction
     PARAMETER (MYN= 1+2*MYE)            ! # nodes in y-direction
     PARAMETER (MTE= MXE*MYE)            ! total # elements
     PARAMETER (MTN= MXN*MYN)            ! total # nodes (quadratic)
     PARAMETER (MTN2= MTN+4*MYN+2*MXN+8) ! tot # nodes (w/4th-order)

     COMMON/GRID/
    *X(MTN2,2),V(MTN,2),PSI(MTN),SURFL(4,MXE),SURFN(4,MXE,2),
    *SURFT(4,MXE,2)

     NSTART= NGNODE-NXP+1
     PSI(NSTART)= 0.
     DO 100 J= NSTART,NGNODE,2
       IF (J.NE.NSTART) THEN

       AX=  2.*V(J,2)-   4.*V(J-1,2)+ 2.*V(J-2,2)
       BX= -3.*V(J-2,2)+ 4.*V(J-1,2)-    V(J,2)
       CX= V(J-2,2)
       AY=  2.*V(J,1)-   4.*V(J-1,1)+ 2.*V(J-2,1)
       BY= -3.*V(J-2,1)+ 4.*V(J-1,1)-    V(J,1)
```

```fortran
          CY= V(J-2,1)

          PSI(J-1)= PSI(J-2)+
     >    (AY/24.+BY/8.+CY/2.)*(X(J,2)-X(J-2,2))-
     >    (AX/24.+BX/8.+CX/2.)*(X(J,1)-X(J-2,1))

          PSI(J)= PSI(J-1)+
     >    (7.*AY/24.+3.*BY/8.+CY/2.)*(X(J,2)-X(J-2,2))-
     >    (7.*AX/24.+3.*BX/8.+CX/2.)*(X(J,1)-X(J-2,1))

          END IF
  100 CONTINUE

      DO 200 J= NSTART,NGNODE
        JT= J-NROW
      JM1= J-NXP
      JM2= J
      DO 200 K= 2,NYP
      AX=  2.*V(JT,2)-  4.*V(JM1,2)+ 2.*V(JM2,2)
      BX= -3.*V(JM2,2)+ 4.*V(JM1,2)-    V(JT,2)
      CX= V(JM2,2)
      AY=  2.*V(JT,1)-  4.*V(JM1,1)+ 2.*V(JM2,1)
      BY= -3.*V(JM2,1)+ 4.*V(JM1,1)-    V(JT,1)
      CY= V(JM2,1)

       PSI(JM1)= PSI(JM2)+
     >  (AY/24.+BY/8.+CY/2.)*(X(JT,2)-X(JM2,2))-
     >  (AX/24.+BX/8.+CX/2.)*(X(JT,1)-X(JM2,1))

       PSI(JT)= PSI(JM1)+
     >  (7.*AY/24.+3.*BY/8.+CY/2.)*(X(JT,2)-X(JM2,2))-
     >  (7.*AX/24.+3.*BX/8.+CX/2.)*(X(JT,1)-X(JM2,1))

      JM2= JT
      JM1= JT-NXP
       JT= JT-NROW

  200 CONTINUE

      RETURN

      END
```

C.   Subroutine GQWEIGHT

```fortran
      SUBROUTINE GQWEIGHT(NPOINT)

C***********************************************************************
C Tabular lookup of abscissae and weight coefficients for gaussian
C quadrature formula. Tables stored in subroutine data blocks.
C Capability of determining coefficients for 2 to 10 point integration.
C***********************************************************************

      COMMON/VINTEG/XI(10,2),W(10)

      DIMENSION C(9,2,10)
```

```fortran
      DATA
     *(C(1,1,J),J=1,10)/-.5773502691, .5773502691, 8*0./,
     *(C(1,2,J),J=1,10)/1.0000000000,1.0000000000, 8*0./
      DATA
     *(C(2,1,J),J=1,10)/-.7745966692,0.0000000000, .7745966692, 7*0./,
     *(C(2,2,J),J=1,10)/ .5555555555, .8888888888, .5555555555, 7*0./
      DATA
     *(C(3,1,J),J=1,10)/-.8611363115,-.3399810435, .3399810435,
     * .8611363115, 6*0./,
     *(C(3,2,J),J=1,10)/ .3478548451, .6521451548, .6521451548,
     * .3478548451, 6*0./
      DATA
     *(C(4,1,J),J=1,10)/-.9061798459,-.5384693101, .0000000000,
     * .5384693101, .9061798459, 5*0./,
     *(C(4,2,J),J=1,10)/ .2369268850, .4786286704, .5688888888,
     * .4786286704, .2369268850, 5*0./
      DATA
     *(C(5,1,J),J=1,10)/-.9324695142,-.6612093864,-.2386191860,
     * .2386191860, .6612093864, .9324695142, 4*0./,
     *(C(5,2,J),J=1,10)/ .1713244923, .3607615730, .4679139345,
     * .4679139345, .3607615730, .1713244923, 4*0./
      DATA
     *(C(6,1,J),J=1,10)/-.9491079123,-.7415311855,-.4058451513,
     *0.0000000000, .4058451513, .7415311855, .9491079123, 3*0./,
     *(C(6,2,J),J=1,10)/ .1294849661, .2797053914, .3818300505,
     * .4179591836, .3818300505, .2797053914, .1294849661, 3*0./
      DATA
     *(C(7,1,J),J=1,10)/-.9602898564,-.7966664774,-.5255324099,
     *-.1834346424, .1834346424, .5255324099, .7966664774,
     * .9602898564, 2*0./,
     *(C(7,2,J),J=1,10)/ .1012285362, .2223810344, .3137066458,
     * .3626837833, .3626837833, .3137066458, .2223810344,
     * .1012285362, 2*0./
      DATA
     *(C(8,1,J),J=1,10)/-.9681602395,-.8360311073,-.6133614327,
     *-.3242534234,0.0000000000, .3242534234, .6133614327,
     * .8360311073, .9681602395, 0./,
     *(C(8,2,J),J=1,10)/ .0812743883, .1806481606, .2606106964,
     * .3123470770, .3302393550, .3123470770, .2606106964,
     * .1806481606, .0812743883, 0./
      DATA
     *(C(9,1,J),J=1,10)/-.9739065285,-.8650633666,-.6794095682,
     *-.4333953941,-.1488743389, .1488743389, .4333953941,
     * .6794095682, .8650633666, .9739065285/,
     *(C(9,2,J),J=1,10)/ .0666713443, .1494513491, .2190863625,
     * .2692667193, .2955242247, .2955242247, .2692667193,
     * .2190863625, .1494513491, .0666713443/

      NINDEX= NPOINT-1

      DO 100 NP= 1,NPOINT
        W(NP)= C(NINDEX,2,NP)
      XI(NP,1)= C(NINDEX,1,NP)
      XI(NP,2)= C(NINDEX,1,NP)
  100 CONTINUE

      RETURN
```

```
        END



D.    Subroutine GEOM

        SUBROUTINE GEOM(NXP,NYP,NXE,NYE,NGNODE,NELEMENT,NCORNER,NROW,NCOL,
     >              IPRESS,NGNODE2,RASPECT,DXE,DYE,DXN,DYN,DXMIN,DYMIN,VOL)

C********************************************************************
C Definition of global node coordinates, global/local coincidence,
C surface nodes, surface/domain coincidence
C********************************************************************
        PARAMETER (MXE= 25,          MYE= 20) ! # elements in x & y direcs
        PARAMETER (MXN= 1+2*MXE)              ! # nodes in x-direction
        PARAMETER (MYN= 1+2*MYE)              ! # nodes in y-direction
        PARAMETER (MTE= MXE*MYE)              ! total # elements
        PARAMETER (MTN= MXN*MYN)              ! total # nodes (quadratic)
        PARAMETER (MTE2= MXE-2+2*(MYE-1))     ! # 6-node pressure elements
        PARAMETER (MTE1= MTE-MTE2-2)          ! # 4-node pressure elements
        PARAMETER (MTC1= (1+MXE)*(1+MYE))     ! # corner nodes
        PARAMETER (MTC2= 2*(1+MYE)+1+MXE+2)   ! extra nodes for mixed press
        PARAMETER (MTC= MTC1+MTC2)            ! total # press nodes
        PARAMETER (MTN2= MTN+4*MYN+2*MXN+8)   ! tot # nodes (w/4th-order)

        COMMON/GRID/
     *XG(MTN2,2),VEL(MTN,2),PSI(MTN),SURFL(4,MXE),SURFN(4,MXE,2),
     *SURFT(4,MXE,2)

        COMMON/GRIDTYPE/
     *CL(MTE1,4),CLLQ(MTE2,6),CLQ(2,9),ECL(MTE,2),CQ(MTE,9),GNODE(MTC),
     *CNODE(MTN),C24(MTE2,15),C44(2,25),PNODE(36),NSURF(4),
     *SURFNODE(4,MXN),CSURF(4,MXE,3),CE(MTN2,4),NCE(MTN2),
     *CES(MXN,2),NCES(MXN),IDIRICH(MTN2)

        COMMON/SURFTYPE/ NESURF(4)

        COMMON/SURFACE/
     *XSURF(MXN),YSURF(MXN),YSURF0(MXN),PSURF(MXN),PBUB(MXN),
     *TSURF(MXN),SURFNN(MXN,2),SURFNT(MXN,2)

        DOUBLE PRECISION XSURF,YSURF,YSURF0,PSURF,TSURF,SURFNN
     >SURFNT,PBUB
        INTEGER CL,CQ,GNODE,CNODE,PNODE,SURFNODE,CSURF,GAM,BETA,R,S,P,ECL,
     >CLLQ,CLQ,CE,CES,C24,C44

        N1= 1
        N2= NXP-2
        NE= 1
        NEL= 1
        NELQ= 1
        NEQ= 1
        NMID= 2*(NXE-2)+1
        DO 2 N= 1,NYE
          DO 3 I= N1,N2,2

          CQ(NE,1)= I+NROW        ! Quadratic coincidence table
```

```
CQ(NE,2)= I+NROW+1
CQ(NE,3)= I+NROW+2
CQ(NE,4)= I+NXP+2
CQ(NE,5)= I+2
CQ(NE,6)= I+1
CQ(NE,7)= I
CQ(NE,8)= I+NXP
CQ(NE,9)= I+NXP+1


IF (N.EQ.1) THEN       ! Linear/mixed coincidence tables
  IF (I.EQ.N1.OR.I.EQ.N2) THEN
      IF (IPRESS.EQ.1) THEN
      CLQ(NEQ,1)= I+NROW
      CLQ(NEQ,2)= I+NROW+1
      CLQ(NEQ,3)= I+NROW+2
      CLQ(NEQ,4)= I+NXP+2
      CLQ(NEQ,5)= I+2
      CLQ(NEQ,6)= I+1
      CLQ(NEQ,7)= I
      CLQ(NEQ,8)= I+NXP
      CLQ(NEQ,9)= I+NXP+1
      ELSE
      CLQ(NEQ,1)= I+NROW
      CLQ(NEQ,2)= I+NROW+2
      CLQ(NEQ,3)= I+2
      CLQ(NEQ,4)= I
      END IF
    IF (I.EQ.N1) THEN
    C44(NEQ,1)=  I+NROW
    C44(NEQ,2)=  NGNODE+25+2*NMID
    C44(NEQ,3)=  I+NROW+1
    C44(NEQ,4)=  C44(NEQ,2)+1
    C44(NEQ,5)=  I+NROW+2
    C44(NEQ,6)=  NGNODE+21+NMID
    C44(NEQ,7)=  I+NXP+2
    C44(NEQ,8)=  NGNODE+9
    C44(NEQ,9)=  I+2
    C44(NEQ,10)= NGNODE+2
    C44(NEQ,11)= I+1
    C44(NEQ,12)= NGNODE+1
    C44(NEQ,13)= I
    C44(NEQ,14)= NGNODE+5
    C44(NEQ,15)= I+NXP
    C44(NEQ,16)= NGNODE+17+NMID
    C44(NEQ,17)= C44(NEQ,16)+1
    C44(NEQ,18)= C44(NEQ,17)+1
    C44(NEQ,19)= C44(NEQ,18)+1
    C44(NEQ,20)= 14+NMID+NGNODE
    C44(NEQ,21)= NGNODE+8
    C44(NEQ,22)= C44(NEQ,21)-1
    C44(NEQ,23)= C44(NEQ,22)-1
    C44(NEQ,24)= 13+NMID+NGNODE
    C44(NEQ,25)= I+NXP+1
    ELSE IF (I.EQ.N2) THEN
    C44(NEQ,1)=  I+NROW
    C44(NEQ,2)=  C44(1,4)+1
    C44(NEQ,3)=  I+NROW+1
    C44(NEQ,4)=  C44(NEQ,2)+1
```

```
          C44(NEQ,5)=  I+NROW+2
          C44(NEQ,6)=  C44(1,6)+3+NMID
          C44(NEQ,7)=  I+NXP+2
          C44(NEQ,8)=  C44(1,8)+3+NMID
          C44(NEQ,9)=  I+2
          C44(NEQ,10)= C44(1,10)+2
          C44(NEQ,11)= I+1
          C44(NEQ,12)= C44(1,10)+1
          C44(NEQ,13)= I
          C44(NEQ,14)= C44(1,14)+3+NMID
          C44(NEQ,15)= I+NXP
          C44(NEQ,16)= C44(1,16)+3+NMID
          C44(NEQ,17)= C44(NEQ,16)+1
          C44(NEQ,18)= C44(NEQ,17)+1
          C44(NEQ,19)= C44(NEQ,18)+1
          C44(NEQ,20)= C44(1,20)+2
          C44(NEQ,21)= C44(NEQ,8)-1
          C44(NEQ,22)= C44(NEQ,21)-1
          C44(NEQ,23)= C44(NEQ,22)-1
          C44(NEQ,24)= C44(1,20)+1
          C44(NEQ,25)= I+NXP+1
          END IF

          ECL(NE,1)= 2
          ECL(NE,2)= NEQ
         NEQ= NEQ+1
      ELSE
          IF (IPRESS.EQ.1) THEN
          CLLQ(NELQ,1)= I+NROW
          CLLQ(NELQ,2)= I+NROW+2
          CLLQ(NELQ,3)= I+NXP+2
          CLLQ(NELQ,4)= I+2
          CLLQ(NELQ,5)= I
          CLLQ(NELQ,6)= I+NXP
          ELSE
          CLLQ(NELQ,1)= I+NROW
          CLLQ(NELQ,2)= I+NROW+2
          CLLQ(NELQ,3)= I+2
          CLLQ(NELQ,4)= I
          END IF

          C24(NELQ,1)=  I+NROW
          C24(NELQ,2)=  C24(NELQ,1)+1
          C24(NELQ,3)=  C24(NELQ,2)+1
          C24(NELQ,4)=  C44(1,6)+(NELQ-1)*2+2
          C24(NELQ,5)=  I+NXP+2
          C24(NELQ,6)=  C44(1,8)+(NELQ-1)*2+2
          C24(NELQ,7)=  I+2
          C24(NELQ,8)=  I+1
          C24(NELQ,9)=  I
          C24(NELQ,10)= C44(1,8)+(NELQ-1)*2
          C24(NELQ,11)= I+NXP
          C24(NELQ,12)= C44(1,6)+(NELQ-1)*2
          C24(NELQ,13)= C24(NELQ,12)+1
          C24(NELQ,14)= I+NXP+1
          C24(NELQ,15)= C24(NELQ,10)+1

          ECL(NE,1)= 1
```

```fortran
              ECL(NE,2)= NELQ
              NELQ= NELQ+1
          END IF
      ELSE
          IF (I.EQ.N1.OR.I.EQ.N2) THEN
              IF (IPRESS.EQ.1) THEN
              CLLQ(NELQ,1)= I+NROW
              CLLQ(NELQ,2)= I+NROW+1
              CLLQ(NELQ,3)= I+NROW+2
              CLLQ(NELQ,4)= I+2
              CLLQ(NELQ,5)= I+1
              CLLQ(NELQ,6)= I
              ELSE
              CLLQ(NELQ,1)= I+NROW
              CLLQ(NELQ,2)= I+NROW+2
              CLLQ(NELQ,3)= I+2
              CLLQ(NELQ,4)= I
              END IF

              NREF= 24+2*NMID+8*(N-1)+1+NGNODE
              IF (I.EQ.N2) NREF= NREF+2
              C24(NELQ,1)=   I+NROW
              C24(NELQ,2)=   NREF
              C24(NELQ,3)=   I+NROW+1
              C24(NELQ,4)=   NREF+1
              C24(NELQ,5)=   I+NROW+2
              C24(NELQ,6)=   I+NXP+2
              C24(NELQ,7)=   I+2
              C24(NELQ,8)=   NREF-7
              C24(NELQ,9)=   I+1
              C24(NELQ,10)= NREF-8
              C24(NELQ,11)= I
              C24(NELQ,12)= I+NXP
              C24(NELQ,13)= NREF-4
              C24(NELQ,14)= I+NXP+1
              C24(NELQ,15)= NREF-3

              ECL(NE,1)= 1
              ECL(NE,2)= NELQ
              NELQ= NELQ+1
          ELSE
              CL(NEL,1)= I+NROW        ! Linear coincidence table
              CL(NEL,2)= I+NROW+2
              CL(NEL,3)= I+2
              CL(NEL,4)= I
              ECL(NE,1)= 0
              ECL(NE,2)= NEL
              NEL= NEL+1
          END IF
      END IF

      NE= NE+1

3   CONTINUE
    N1= N1+NROW
    N2= N2+NROW
2 CONTINUE
    NEQ= NEQ-1
```

```
      NELQ= NELQ-1

C Corner node identifier

      DO 967 N= 1,NGNODE
  967 CNODE(N)= 0
      N1= 1
      N2= NXP
      NCORNER= 0
      DO 4 I= 1,NYP,2
        DO 5 K= N1,N2,2
      NCORNER= NCORNER+1
      CNODE(K)= NCORNER
    5   GNODE(NCORNER)= K
        N1= N1+NROW
        N2= N2+NROW
    4 CONTINUE

      IF (IPRESS.EQ.1) THEN
      DO 30 N= 1,2
      IF (N.EQ.1) K= 2
      IF (N.EQ.2) K= NXP-1
      DO 30 I= 1,NYE+1
        NCORNER= NCORNER+1
        CNODE(K)= NCORNER
        GNODE(NCORNER)= K
        K= K+NROW
   30 CONTINUE
      DO 31 K= NXP+1,2*NXP,2
      NCORNER= NCORNER+1
      CNODE(K)= NCORNER
      GNODE(NCORNER)= K
   31 CONTINUE
      CNODE(NXP+2)= NCORNER+1
      GNODE(NCORNER+1)= NXP+2
      CNODE(2*NXP-1)= NCORNER+2
      GNODE(NCORNER+2)= 2*NXP-1
      NCORNER= NCORNER+2
      END IF

C SET GLOBAL COORDINATES

      DO 6 I= 1,NYE+1
      NS= (I-1)*NROW+1
      N= NS
      DO 6 J= 1,NXP
        XG(N,1)= REAL(XSURF(J))
        XG(N,2)= XG(NS,2)*REAL(YSURF(J))/REAL(YSURF(1))
        N= N+1
    6 CONTINUE

      NE= 0
      DO 7 I= 1,NYE
        DO 8 J= 1,NXE
        NE= NE+1
        DO 8 K= 1,2
          IF (I.EQ.NYE) XG(CQ(NE,2),K)=
    *       (XG(CQ(NE,1),K)+XG(CQ(NE,3),K))/2.
```

```fortran
            XG(CQ(NE,8),K)=(XG(CQ(NE,7),K)+XG(CQ(NE,1),K))/2.
   8    CONTINUE
        DO 9 K= 1,2
   9    XG(CQ(NE,4),K)=(XG(CQ(NE,3),K)+XG(CQ(NE,5),K))/2.
   7  CONTINUE
        DO 10 NE= 1,NELEMENT
        XG(CQ(NE,9),1)=(XG(CQ(NE,2),1)+XG(CQ(NE,6),1))/2.
        XG(CQ(NE,9),2)=(XG(CQ(NE,2),2)+XG(CQ(NE,6),2))/2.
  10  CONTINUE

C x-coordinates of quadratic node additions

        DO 100 N= 1,NEQ
        XG(C44(N,12),1)= .5*(XG(C44(N,13),1)+XG(C44(N,11),1))
        XG(C44(N,23),1)= XG(C44(N,12),1)
        XG(C44(N,24),1)= XG(C44(N,12),1)
        XG(C44(N,17),1)= XG(C44(N,12),1)
        XG(C44(N, 2),1)= XG(C44(N,12),1)
        XG(C44(N,10),1)= .5*(XG(C44(N,11),1)+XG(C44(N, 9),1))
        XG(C44(N,21),1)= XG(C44(N,10),1)
        XG(C44(N,20),1)= XG(C44(N,10),1)
        XG(C44(N,19),1)= XG(C44(N,10),1)
        XG(C44(N, 4),1)= XG(C44(N,10),1)
        XG(C44(N,22),1)= XG(C44(N,11),1)
        XG(C44(N,18),1)= XG(C44(N,11),1)
        XG(C44(N,14),1)= XG(C44(N,13),1)
        XG(C44(N,16),1)= XG(C44(N,13),1)
        XG(C44(N, 8),1)= XG(C44(N, 9),1)
        XG(C44(N, 6),1)= XG(C44(N, 9),1)
 100  CONTINUE
        DO 101 N= 1,NELQ
        IF (N.LE.NXE-2) THEN
           XG(C24(N,10),1)= XG(C24(N, 9),1)
           XG(C24(N,12),1)= XG(C24(N, 9),1)
           XG(C24(N,15),1)= XG(C24(N, 8),1)
           XG(C24(N,13),1)= XG(C24(N, 8),1)
           XG(C24(N, 6),1)= XG(C24(N, 7),1)
           XG(C24(N, 4),1)= XG(C24(N, 7),1)
        ELSE
           XG(C24(N,10),1)= .5*(XG(C24(N,11),1)+XG(C24(N, 9),1))
           XG(C24(N, 8),1)= .5*(XG(C24(N, 9),1)+XG(C24(N, 7),1))
           XG(C24(N,13),1)= XG(C24(N,10),1)
           XG(C24(N, 2),1)= XG(C24(N,10),1)
           XG(C24(N,15),1)= XG(C24(N, 8),1)
           XG(C24(N, 4),1)= XG(C24(N, 8),1)
        END IF
 101  CONTINUE
        DO 110 N= 1,NEQ
        XG(C44(N,12),2)= QCAL(XG(C44(N,12),1),
     >      XG(C44(N,13),1),XG(C44(N,11),1),XG(C44(N, 9),1),
     >      XG(C44(N,13),2),XG(C44(N,11),2),XG(C44(N, 9),2))
        XG(C44(N,10),2)= QCAL(XG(C44(N,10),1),
     >      XG(C44(N,13),1),XG(C44(N,11),1),XG(C44(N, 9),1),
     >      XG(C44(N,13),2),XG(C44(N,11),2),XG(C44(N, 9),2))
        XG(C44(N,24),2)= QCAL(XG(C44(N,24),1),
     >      XG(C44(N,15),1),XG(C44(N,25),1),XG(C44(N, 7),1),
     >      XG(C44(N,15),2),XG(C44(N,25),2),XG(C44(N, 7),2))
        XG(C44(N,20),2)= QCAL(XG(C44(N,20),1),
```

```
     >      XG(C44(N,15),1),XG(C44(N,25),1),XG(C44(N, 7),1),
     >      XG(C44(N,15),2),XG(C44(N,25),2),XG(C44(N, 7),2))
        XG(C44(N, 2),2)= QCAL(XG(C44(N, 2),1),
     >      XG(C44(N, 1),1),XG(C44(N, 3),1),XG(C44(N, 5),1),
     >      XG(C44(N, 1),2),XG(C44(N, 3),2),XG(C44(N, 5),2))
        XG(C44(N, 4),2)= QCAL(XG(C44(N, 4),1),
     >      XG(C44(N, 1),1),XG(C44(N, 3),1),XG(C44(N, 5),1),
     >      XG(C44(N, 1),2),XG(C44(N, 3),2),XG(C44(N, 5),2))
        XG(C44(N,14),2)= .5*(XG(C44(N,13),2)+XG(C44(N,15),2))
        XG(C44(N,16),2)= .5*(XG(C44(N,15),2)+XG(C44(N, 1),2))
        XG(C44(N,23),2)= .5*(XG(C44(N,12),2)+XG(C44(N,24),2))
        XG(C44(N,17),2)= .5*(XG(C44(N,24),2)+XG(C44(N, 2),2))
        XG(C44(N,22),2)= .5*(XG(C44(N,11),2)+XG(C44(N,25),2))
        XG(C44(N,18),2)= .5*(XG(C44(N,25),2)+XG(C44(N, 3),2))
        XG(C44(N,21),2)= .5*(XG(C44(N,10),2)+XG(C44(N,20),2))
        XG(C44(N,19),2)= .5*(XG(C44(N,20),2)+XG(C44(N, 4),2))
        XG(C44(N, 8),2)= .5*(XG(C44(N, 9),2)+XG(C44(N, 7),2))
        XG(C44(N, 6),2)= .5*(XG(C44(N, 7),2)+XG(C44(N, 5),2))
 110 CONTINUE
     DO 111 N= 1,NELQ
     IF (N.LE.NXE-2) THEN
        XG(C24(N,15),2)= .5*(XG(C24(N, 8),2)+XG(C24(N,14),2))
        XG(C24(N,13),2)= .5*(XG(C24(N,14),2)+XG(C24(N, 2),2))
        XG(C24(N, 6),2)= .5*(XG(C24(N, 7),2)+XG(C24(N, 5),2))
        XG(C24(N, 4),2)= .5*(XG(C24(N, 5),2)+XG(C24(N, 3),2))
     ELSE
        XG(C24(N, 2),2)= QCAL(XG(C24(N, 2),1),
     >      XG(C24(N, 1),1),XG(C24(N, 3),1),XG(C24(N, 5),1),
     >      XG(C24(N, 1),2),XG(C24(N, 3),2),XG(C24(N, 5),2))
        XG(C24(N, 4),2)= QCAL(XG(C24(N, 4),1),
     >      XG(C24(N, 1),1),XG(C24(N, 3),1),XG(C24(N, 5),1),
     >      XG(C24(N, 1),2),XG(C24(N, 3),2),XG(C24(N, 5),2))
        XG(C24(N,13),2)= .5*(XG(C24(N,10),2)+XG(C24(N, 2),2))
        XG(C24(N,15),2)= .5*(XG(C24(N, 8),2)+XG(C24(N, 4),2))
     END IF
 111 CONTINUE

C Minimum grid dimensions

     DXMIN= XG(2,1)-XG(1,1)
     DYMIN= XG(NXE+1,2)-XG(NXE+1+NROW,2)

C Boundary node definition and type

     NSURF(1)= NYP
     NSURF(3)= NYP
     NSURF(2)= NXP-2
     NSURF(4)= NXP-2

     NP= -1*NXP+1
     DO 11 N= 1,NSURF(1)
       NP=NP+NXP
       SURFNODE(1,N)= NP
  11 CONTINUE

     NP= NGNODE-NXP+1
     DO 12 N= 1,NSURF(2)
       NP=NP+1
```

```
              SURFNODE(2,N)= NP
   12 CONTINUE

      NP= 0
      DO 13 N= 1,NSURF(3)
         NP=NP+NXP
      SURFNODE(3,N)= NP
   13 CONTINUE

      NP= 1
      DO 14 N= 1,NSURF(4)
         NP= NP+1
      SURFNODE(4,N)= NP
   14 CONTINUE

C Boundary element definition and properties

      NESURF(1)= NYE
      NESURF(3)= NYE
      NESURF(2)= NXE
      NESURF(4)= NXE

      NG= 1
      DO 15 N= 1,NESURF(1)
         CSURF(1,N,1)= NG+NROW
      CSURF(1,N,2)= NG+NXP
      CSURF(1,N,3)= NG
      SURFL(1,N)= SQRT((XG(NG+NROW,1)-XG(NG,1))**2+
     *                  (XG(NG+NROW,2)-XG(NG,2))**2)
      SURFT(1,N,1)= (XG(NG,1)-XG(NG+NROW,1))/SURFL(1,N)
      SURFT(1,N,2)= (XG(NG,2)-XG(NG+NROW,2))/SURFL(1,N)
      SURFN(1,N,1)= -1.*SURFT(1,N,2)
      SURFN(1,N,2)=     SURFT(1,N,1)
         NG= NG+NROW
   15 CONTINUE

      NG= NGNODE
      DO 18 N= 1,NESURF(2)
         CSURF(2,N,1)= NG
      CSURF(2,N,2)= NG-1
      CSURF(2,N,3)= NG-2
      SURFL(2,N)= SQRT((XG(NG-2,1)-XG(NG,1))**2+
     *                  (XG(NG-2,2)-XG(NG,2))**2)
      SURFT(2,N,1)= (XG(NG-2,1)-XG(NG,1))/SURFL(2,N)
      SURFT(2,N,2)= (XG(NG-2,2)-XG(NG,2))/SURFL(2,N)
      SURFN(2,N,1)= -1.*SURFT(2,N,2)
      SURFN(2,N,2)=     SURFT(2,N,1)
         NG= NG-2
   18 CONTINUE

      NG= NXP
      DO 16 N= 1,NESURF(3)
         CSURF(3,N,1)= NG
      CSURF(3,N,2)= NG+NXP
      CSURF(3,N,3)= NG+NROW
      SURFL(3,N)= SQRT((XG(NG+NROW,1)-XG(NG,1))**2+
     *                  (XG(NG+NROW,2)-XG(NG,2))**2)
      SURFT(3,N,1)= (XG(NG+NROW,1)-XG(NG,1))/SURFL(3,N)
```

```
      SURFT(3,N,2)= (XG(NG+NROW,2)-XG(NG,2))/SURFL(3,N)
      SURFN(3,N,1)= -1.*SURFT(3,N,2)
      SURFN(3,N,2)=     SURFT(3,N,1)
        NG= NG+NROW
  16 CONTINUE

      VOL= 0.
      NG= 1
      DO 17 N= 1,NESURF(4)
        CSURF(4,N,1)= NG
      CSURF(4,N,2)= NG+1
      CSURF(4,N,3)= NG+2
      X1= XG(NG,1)
      X2= XG(NG+1,1)
      X3= XG(NG+2,1)
      V1= XG(NG,2)
      V2= XG(NG+1,2)
      V3= XG(NG+2,2)
      A= ((V1-V2)/(X1-X2)-(V2-V3)/(X2-X3))/(X1-X3)
      B= (V1-V2)/(X1-X2)-A*(X1+X2)
      C= V1-A*X1**2-B*X1
      X1P= 2.*A*X1+B
      X3P= 2.*A*X3+B
      SURFL(4,N)= (X3P*SQRT(X3P**2+1.)+ALOG(X3P+SQRT(X3P**2+1.))-
     *           X1P*SQRT(X1P**2+1.)-ALOG(X1P+SQRT(X1P**2+1.)))/
     *           4./A
      SURFT(4,N,1)= (XG(NG+2,1)-XG(NG,1))/SURFL(4,N)
      SURFT(4,N,2)= (XG(NG+2,2)-XG(NG,2))/SURFL(4,N)
      SURFN(4,N,1)= -1.*SURFT(4,N,2)
      SURFN(4,N,2)=     SURFT(4,N,1)
      VOL= VOL+.5*(XG(NG+2,2)+XG(NG,2))*(XG(NG+2,1)-XG(NG,1))
        NG= NG+2
  17 CONTINUE

C Node-Element Correlation Table

      DO 35 NG= 1,NGNODE2
      NCE(NG)= 0
      IDIRICH(NG)= 0
      DO 35 N= 1,4
        CE(NG,N)= 0
  35 CONTINUE

      DO 40 NE= 1,NELEMENT
      IF (ECL(NE,1).EQ.0) THEN
        NM= 9
      ELSE IF (ECL(NE,1).EQ.1) THEN
        NM= 15
      ELSE IF (ECL(NE,1).EQ.2) THEN
        NM= 25
      END IF
      DO 40 N= 1,NM
        IF (ECL(NE,1).EQ.0) THEN
          NG= CQ(NE,N)
        ELSE IF (ECL(NE,1).EQ.1) THEN
          NG= C24(ECL(NE,2),N)
        ELSE IF (ECL(NE,1).EQ.2) THEN
          NG= C44(ECL(NE,2),N)
```

```
          END IF
          IF (NCE(NG).EQ.0) THEN
            NCE(NG)= NCE(NG)+1
            CE(NG,NCE(NG))= NE
          ELSE
            DO 41 NC= 1,NCE(NG)
41            IF (CE(NG,NC).EQ.NE) GOTO 42
            NCE(NG)= NCE(NG)+1
            CE(NG,NCE(NG))= NE
42          CONTINUE
          END IF
40 CONTINUE

      DO 50 NG= 1,NXP
      NCES(NG)= 0
      DO 50 N= 1,2
        CES(NG,N)= 0
50 CONTINUE

      DO 60 NE= 1,NESURF(4)
      DO 60 N= 1,3
        NG= CSURF(4,NE,N)
        IF (NCES(NG).EQ.0) THEN
          NCES(NG)= NCES(NG)+1
          CES(NG,NCES(NG))= NE
        ELSE
          DO 61 NC= 1,NCES(NG)
61          IF (CES(NG,NC).EQ.NE) GOTO 62
          NCES(NG)= NCES(NG)+1
          CES(NG,NCES(NG))= NE
62        CONTINUE
        END IF
60 CONTINUE

      NL= 1
      NR= NXP
      DO 80 N= 1,NYP
      IDIRICH(NL)= 1
      IDIRICH(NR)= 1           -
      NL= NL+NXP
      NR= NR+NXP
80 CONTINUE
      DO 81 N= NGNODE-NXP+1,NGNODE
81 IDIRICH(N)= 1
      IDIRICH(NGNODE+5)= 1
      IDIRICH(NGNODE+12+NMID)= 1
      IDIRICH(NGNODE+17+NMID)= 1
      IDIRICH(NGNODE+24+2*NMID)= 1

      RETURN
      END
```

E.   Function QCAL

```
      FUNCTION QCAL(X,X1,X2,X3,V1,V2,V3)
```

```
C*******************************************************************
C Calculates y-coord along 2nd-order curve
C*******************************************************************
      A= ((V1-V2)/(X1-X2)-(V2-V3)/(X2-X3))/(X1-X3)
      B= (V1-V2)/(X1-X2)-A*(X1+X2)
      C= V1-A*X1**2-B*X1

      QCAL= A*X**2+B*X+C

      RETURN
      END
```

F.   Subroutine GEOM2

```
      SUBROUTINE GEOM2(NXE,NYE,NXP,NROW,NSMALL,RASPECT,FSCALE)

      PARAMETER (MXE= 25,        MYE= 20) ! # elements in x & y direcs
      PARAMETER (MXN= 1+2*MXE)            ! # nodes in x-direction
      PARAMETER (MYN= 1+2*MYE)            ! # nodes in y-direction
      PARAMETER (MTE= MXE*MYE)            ! total # elements
      PARAMETER (MTN= MXN*MYN)            ! total # nodes (quadratic)
      PARAMETER (MTN2= MTN+4*MYN+2*MXN+8) ! tot # nodes (w/4th-order)

      COMMON/GRID/
     *XG(MTN2,2),VEL(MTN,2),PSI(MTN),SURFL(4,MXE),SURFN(4,MXE,2),
     *SURFT(4,MXE,2)

      COMMON/SURFACE/
     *XSURF(MXN),YSURF(MXN),YSURF0(MXN),PSURF(MXN),PBUB(MXN),
     *TSURF(MXN),SURFNN(MXN,2),SURFNT(MXN,2)

      DIMENSION FSCALE(2),NSMALL(2)
      DOUBLE PRECISION XSURF,YSURF,YSURF0,PSURF,TSURF,SURFNN
     >SURFNT,PBUB

      IF (NSMALL(1).EQ.0) FSCALE(1)= 1.
      IF (NSMALL(2).EQ.0) FSCALE(2)= 1.
      DXS= FSCALE(1)*REAL(NXE)+REAL(NSMALL(1))*(1.-FSCALE(1))
      DYS= FSCALE(2)*REAL(NYE)+REAL(NSMALL(2))*(1.-FSCALE(2))
      DXS= 1./DXS/2.
      DYS= 1./DYS/2.
      DXL= FSCALE(1)*DXS
      DYL= FSCALE(2)*DYS
      NSX= NSMALL(1)
      NSY= 2*NSMALL(2)
      NLX= NXE-NSX
      NLY= NYE*2-NSY

      XSURF(1)= 0.
      XSURF(NXP)= 1.

      DO 3 N= 2,NSX+NLX+1
      NN= NXP-N+1
      IF (N-1.LE.NSX) THEN
        DX= DXS
```

201

```
      ELSE
         DX= DXL
      END IF
      XSURF(N)= XSURF(N-1)+DX
      XSURF(NN)= XSURF(NN+1)-DX
   3 CONTINUE
      XSURF(NXE+1)= 0.5

      XG(1,2)= RASPECT
      NP= 1
      DO 4 N= 2,NSY+NLY+1
      NP= NP+NXP
      IF (N-1.LE.NSY) THEN
         DY= DYS
      ELSE
         DY= DYL
      END IF
      XG(NP,2)= XG(NP-NXP,2)-DY
   4 CONTINUE
      XG(NP,2)= 0.

      RETURN
      END
```

G.   Subroutine TENSCAL

```
      SUBROUTINE TENSCAL(NELEMENT,NLNQ,NLNL,NPOINT,IPRESS,PR,PVAP)

C******************************************************************************
C Calculates tensors used in formulation of finite element equations.
C******************************************************************************

      PARAMETER (MXE= 25,           MYE= 20) ! # elements in x & y direcs
      PARAMETER (MXN= 1+2*MXE)                ! # nodes in x-direction
      PARAMETER (MYN= 1+2*MYE)                ! # nodes in y-direction
      PARAMETER (MTE= MXE*MYE)                ! total # elements
      PARAMETER (MTN= MXN*MYN)                ! total # nodes (quadratic)
      PARAMETER (MTE2= MXE-2+2*(MYE-1))       ! # 6-node pressure elements
      PARAMETER (MTE1= MTE-MTE2-2)            ! # 4-node pressure elements
      PARAMETER (MTC1= (1+MXE)*(1+MYE))       ! # corner nodes
      PARAMETER (MTC2= 2*(1+MYE)+1+MXE+2)     ! extra nodes for mixed press
      PARAMETER (MTC= MTC1+MTC2)              ! total # press nodes
      PARAMETER (MTN2= MTN+4*MYN+2*MXN+8)     ! tot # nodes (w/4th-order)

      COMMON/VINTEG/XI(10,2),W(10)

      COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

      COMMON/VQUAD/
     *A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     *XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

      COMMON/GRID/
     *XG(MTN2,2),VEL(MTN,2),PSI(MTN),SURFL(4,MXE),SURFN(4,MXE,2),
     *SURFT(4,MXE,2)
```

```
      COMMON/GRIDTYPE/
*CL(MTE1,4),CLLQ(MTE2,6),CLQ(2,9),ECL(MTE,2),CQ(MTE,9),GNODE(MTC),
*CNODE(MTN),C24(MTE2,15),C44(2,25),PNODE(36),NSURF(4),
*SURFNODE(4,MXN),CSURF(4,MXE,3),CE(MTN2,4),NCE(MTN2),
*CES(MXN,2),NCES(MXN),IDIRICH(MTN2)

      COMMON/SURFTYPE/ NESURF(4)

      COMMON/SURFACE/
*XSURF(MXN),YSURF(MXN),YSURF0(MXN),PSURF(MXN),PBUB(MXN),
*TSURF(MXN),SURFNN(MXN,2),SURFNT(MXN,2)

      COMMON/TENSOR/
*EA(MTE,9,9),        EB(MTE,9,9,9,2),       EC(MTE,9,9),
*EA24V(MTE2,9,15),   EB24(MTE2,15,9,15,2),  EC24(MTE2,15,15),
*EA44V(2,9,25),      EB44(2,25,9,25,2),     EC44(2,25,25),
*EA24T(MTE2,15,15),  EA44T(2,25,25),
*ED(MTE1,9,4,2),     EDLQ(MTE2,9,6,2),      EDQ(2,9,9,2),
*EE(MTE1,4,4),       EELQ(MTE2,6,6),        EEQ(2,9,9),
*EF(MTE1,4,9,2),     EFLQ(MTE2,6,9,2),      EFQ(2,9,9,2),
*EG(MTE1,4,9,2,9,2),EGLQ(MTE2,6,9,2,9,2), EGQ(2,9,9,2,9,2),
*EH(MTE1,4,9,9,2,2),EHLQ(MTE2,6,9,9,2,2), EHQ(2,9,9,9,2,2),
*EQ(MTE,9,2,9,2),    SAQ(MXE,3,9),          SAQ2(2,5,25)

      DOUBLE PRECISION XSURF,YSURF,YSURF0,PSURF,TSURF,SURFNN,
>SURFNT,PBUB
      INTEGER CL,CQ,CLQ,CLLQ,GNODE,CNODE,PNODE,SURFNODE,CSURF,ECL
      INTEGER GAM,BETA,R,S,P,CE,CES,C24,C44
      REAL JINV

C EA          Momentum, Energy, Vel Corr
C EB          Momentum, Energy
C EC          Momentum, Energy
C ED          Momentum, Vel Corr
C EE          Press Corr init and final
C EF          Press Corr final
C EG          Press Corr final
C EH          Press Corr final
C EQ          Momentum

C Natural element coordinates: 4-node linear representation

      XINL(1,1)=-1.
      XINL(2,1)= 1.
      XINL(3,1)= 1.
      XINL(4,1)=-1.
      XINL(1,2)=-1.
      XINL(2,2)=-1.
      XINL(3,2)= 1.
      XINL(4,2)= 1.

C Natural element coordinates: 6-node mixed linear/quad representation

      XINLQ(1,1,1)=-1.        ! Side element
      XINLQ(2,1,1)= 0.
      XINLQ(3,1,1)= 1.
      XINLQ(4,1,1)= 1.
      XINLQ(5,1,1)= 0.
```

```fortran
      XINLQ(6,1,1)=-1.
      XINLQ(1,2,1)=-1.
      XINLQ(2,2,1)=-1.
      XINLQ(3,2,1)=-1.
      XINLQ(4,2,1)= 1.
      XINLQ(5,2,1)= 1.
      XINLQ(6,2,1)= 1.

      XINLQ(1,1,2)=-1.          ! Top element
      XINLQ(2,1,2)= 1.
      XINLQ(3,1,2)= 1.
      XINLQ(4,1,2)= 1.
      XINLQ(5,1,2)=-1.
      XINLQ(6,1,2)=-1.
      XINLQ(1,2,2)=-1.
      XINLQ(2,2,2)=-1.
      XINLQ(3,2,2)= 0.
      XINLQ(4,2,2)= 1.
      XINLQ(5,2,2)= 1.
      XINLQ(6,2,2)= 0.

C Natural element coordinates: 9-node quadratic representation

      XINQ(1,1)=-1.
      XINQ(2,1)= 0.
      XINQ(3,1)= 1.
      XINQ(4,1)= 1.
      XINQ(5,1)= 1.
      XINQ(6,1)= 0.
      XINQ(7,1)=-1.
      XINQ(8,1)=-1.
      XINQ(9,1)= 0.
      XINQ(1,2)=-1.
      XINQ(2,2)=-1.
      XINQ(3,2)=-1.
      XINQ(4,2)= 0.
      XINQ(5,2)= 1.
      XINQ(6,2)= 1.
      XINQ(7,2)= 1.
      XINQ(8,2)= 0.
      XINQ(9,2)= 0.

C*********************************************************************
C GAUSSIAN QUADRATURE PARAMETERS
C*********************************************************************

C Determine abscissae and weight coefficients for specified number of Gaussian
C Quadrature integration points. Subroutine employs tabular lookup.

      CALL GQWEIGHT(NPOINT)

C Clear array values

      DO 20 NE= 1,NELEMENT

        DO 21 N= 1,NLNQ
        DO 21 M= 1,NLNQ
          EA(NE,N,M)= 0.
```

```
            EC(NE,N,M)= 0.
            DO 21 I= 1,2
               DO 23 J= 1,2
23                EQ(NE,N,J,M,I)= 0.
               DO 21 GAM= 1,NLNQ
                  EB(NE,N,GAM,M,I)= 0.
21       CONTINUE

      IF (ECL(NE,1).EQ.0) THEN
        NLN= 4
        NTN= 9
      ELSE IF (ECL(NE,1).EQ.1) THEN
        NLN= 6
        NTN= 15
      ELSE IF (ECL(NE,1).EQ.2) THEN
        NLN= 9
        NTN= 25
      END IF

        IF (NTN.NE.9) THEN
        DO 60 N= 1,NTN
        DO 61 M= 1,NLNQ
          IF (NTN.EQ.15) THEN
            EA24V(ECL(NE,2),M,N)= 0.
          ELSE
            EA44V(ECL(NE,2),M,N)= 0.
          END IF
          DO 61 L= 1,NTN
          DO 61 I= 1,2
            IF (NTN.EQ.15) THEN
            EB24(ECL(NE,2),N,M,L,I)= 0.
            ELSE
            EB44(ECL(NE,2),N,M,L,I)= 0.
            END IF
61      CONTINUE
        DO 60 M= 1,NTN
          IF (NTN.EQ.15) THEN
            EA24T(ECL(NE,2),N,M)= 0.
            EC24(ECL(NE,2),N,M)= 0.
          ELSE
            EA44T(ECL(NE,2),N,M)= 0.
            EC44(ECL(NE,2),N,M)= 0.
          END IF
60    CONTINUE
      END IF

        DO 22 N= 1,NLNQ
        DO 22 M= 1,NLN
        DO 22 I= 1,2
          IF (ECL(NE,1).EQ.0) THEN
            ED(ECL(NE,2),N,M,I)= 0.
            EF(ECL(NE,2),M,N,I)= 0.
          ELSE IF (ECL(NE,1).EQ.1) THEN
            EDLQ(ECL(NE,2),N,M,I)= 0.
            EFLQ(ECL(NE,2),M,N,I)= 0.
          ELSE IF (ECL(NE,1).EQ.2) THEN
            EDQ(ECL(NE,2),N,M,I)= 0.
            EFQ(ECL(NE,2),M,N,I)= 0.
```

```
          END IF
          DO 22 GAM= 1,NLNQ
          DO 22 J= 1,2
            IF (ECL(NE,1).EQ.0) THEN
              EG(ECL(NE,2),M,N,I,GAM,J)= 0.
              EH(ECL(NE,2),M,N,GAM,I,J)= 0.
            ELSE IF (ECL(NE,1).EQ.1) THEN
              EGLQ(ECL(NE,2),M,N,I,GAM,J)= 0.
              EHLQ(ECL(NE,2),M,N,GAM,I,J)= 0.
            ELSE
              EGQ(ECL(NE,2),M,N,I,GAM,J)= 0.
              EHQ(ECL(NE,2),M,N,GAM,I,J)= 0.
            END IF
    22 CONTINUE
C
          DO 20 N= 1,NLN
          DO 20 M= 1,NLN
            IF (ECL(NE,1).EQ.0) THEN
              EE(ECL(NE,2),N,M)= 0.
            ELSE IF (ECL(NE,1).EQ.1) THEN
              EELQ(ECL(NE,2),N,M)= 0.
            ELSE
              EEQ(ECL(NE,2),N,M)= 0.
            END IF
C
    20 CONTINUE

          DO 29 NP= 1,NESURF(4)
          DO 29 N= 1,3
          DO 29 M= 1,NLNQ
          SAQ(NP,N,M)= 0.
    29 CONTINUE

          DO 89 NP= 1,2
          DO 89 N= 1,5
          DO 89 M= 1,25
          SAQ2(NP,N,M)= 0.
    89 CONTINUE

C Isoparametric function calculation loop

          DO 30 NE= 1,NELEMENT

          NLN= 4
          IF (ECL(NE,1).EQ.0) THEN
            NLN= 4
            NTN= 9
          ELSE IF (ECL(NE,1).EQ.1) THEN
            IF (IPRESS.EQ.1) NLN= 6
            NTN= 15
          ELSE IF (ECL(NE,1).EQ.2) THEN
            IF (IPRESS.EQ.1) NLN= 9
            NTN= 25
          END IF

C Coordinate coefficients

          DO 31 N= 1,2
```

206

```
C
C Quadratic interpolation function (corners and midnodes)
C
          A2(N)=   4.*XG(CQ(NE,9),N)
          B2(N)=   2.*XG(CQ(NE,4),N)-2.*XG(CQ(NE,8),N)
          C2(N)=  -2.*XG(CQ(NE,2),N)+2.*XG(CQ(NE,6),N)
          D2(N)=      XG(CQ(NE,1),N)-    XG(CQ(NE,3),N)
     *             +XG(CQ(NE,5),N)-    XG(CQ(NE,7),N)
          E2(N)=   2.*XG(CQ(NE,4),N)+2.*XG(CQ(NE,8),N)
     *             -4.*XG(CQ(NE,9),N)
          F2(N)=   2.*XG(CQ(NE,2),N)+2.*XG(CQ(NE,6),N)
     *             -4.*XG(CQ(NE,9),N)
          G2(N)=  -1.*XG(CQ(NE,1),N)+    XG(CQ(NE,3),N)
     *             +XG(CQ(NE,5),N)-    XG(CQ(NE,7),N)
     *             -2.*XG(CQ(NE,4),N)+2.*XG(CQ(NE,8),N)
          H2(N)=  -1.*XG(CQ(NE,1),N)-    XG(CQ(NE,3),N)
     *             +XG(CQ(NE,5),N)+    XG(CQ(NE,7),N)
     *             +2.*XG(CQ(NE,2),N)-2.*XG(CQ(NE,6),N)
          P2(N)=      XG(CQ(NE,1),N)+    XG(CQ(NE,3),N)
     *             +XG(CQ(NE,5),N)+    XG(CQ(NE,7),N)
     *             -2.*XG(CQ(NE,2),N)-2.*XG(CQ(NE,4),N)
     *             -2.*XG(CQ(NE,6),N)-2.*XG(CQ(NE,8),N)
     *             +4.*XG(CQ(NE,9),N)
C
   31 CONTINUE
C
      DO 30 R= 1,NPOINT
      DO 30 S= 1,NPOINT
C
      ET1= XI(R,1)
      ET2= XI(S,2)
C
C Coefficients used in Jacobian and derivative calculations
C
      DO 38 P= 1,2
        A(P)= B2(P)+D2(P)*ET2+2.*E2(P)*ET1+G2(P)*ET2**2
     *        +2.*H2(P)*ET1*ET2 +2.*P2(P)*ET1*ET2**2
        B(P)= C2(P)+D2(P)*ET1+2.*F2(P)*ET2+H2(P)*ET1**2
     *        +2.*G2(P)*ET1*ET2 +2.*P2(P)*ET2*ET1**2
        AD1(P)= 2.*(E2(P)+H2(P)*ET2+P2(P)*ET2**2)
        AD2(P)= D2(P)+2.*G2(P)*ET2+2.*H2(P)*ET1+4.*P2(P)*ET1*ET2
        BD1(P)= AD2(P)
        BD2(P)= 2.*(F2(P)+G2(P)*ET1+P2(P)*ET1**2)
   38 CONTINUE
C
C Jacobian determinant: quadratic variation
C
      DETJQ= 0.
      DO 33 N= 1,2
      DO 33 M= 1,2
        IF (M.EQ.N) GOTO 33
        DETJQ= EPS(M,N)/16.*A(M)*B(N)+DETJQ
   33 CONTINUE
C
      WRSJQ= W(R)*W(S)*DETJQ
C
C Inverse Jacobian tensor
C
```

```
      DO 34 I= 1,2
      DO 34 K=.1,2
        IF (I.EQ.1) THEN
        P= 2
        SIGN= 1.
        ELSE
        P= 1
        SIGN= -1.
      END IF
      IF (K.EQ.1) THEN
        JINV(I,K)= SIGN/4./DETJQ*B(P)
      ELSE
        JINV(I,K)= -1.*SIGN/4./DETJQ*A(P)
      END IF
   34 CONTINUE
C
C Coefficient Tensors
C
C Quadratic isoparametric functions only
C
      DO 35 N= 1,NLNQ
      DO 35 M= 1,NLNQ
C
      EA(NE,N,M)= EA(NE,N,M)+ WRSJQ*PHIQ(N,ET1,ET2)*PHIQ(M,ET1,ET2)
C
      SI= 0.
      DO 36 I= 1,2
   36 SI=SI+DPDXQ(N,I,ET1,ET2,NE)*DPDXQ(M,I,ET1,ET2,NE)
      EC(NE,N,M)= EC(NE,N,M)+ WRSJQ*SI
C
      DO 35 I= 1,2

      DO 37 J= 1,2
   37   EQ(NE,N,J,M,I)= EQ(NE,N,J,M,I)+WRSJQ*DPDXQ(N,J,ET1,ET2,NE)*
     >                    DPDXQ(M,I,ET1,ET2,NE)

      DO 35 L= 1,NLNQ
C
        EB(NE,N,L,M,I)= EB(NE,N,L,M,I)+
     *  WRSJQ*PHIQ(N,ET1,ET2)*PHIQ(L,ET1,ET2)*DPDXQ(M,I,ET1,ET2,NE)
C
   35 CONTINUE


C 4th-order isoparametric functions only

      IF (NTN.NE.9) THEN
      DO 70 N= 1,NTN

      DO 274 M= 1,NLNQ
        IF (NTN.EQ.15) THEN
          EA24V(ECL(NE,2),M,N)= EA24V(ECL(NE,2),M,N)+
     >      WRSJQ*PHIQ(M,ET1,ET2)*PHI24(N,ET1,ET2,NE)
        ELSE
          EA44V(ECL(NE,2),M,N)= EA44V(ECL(NE,2),M,N)+
     >      WRSJQ*PHIQ(M,ET1,ET2)*PHI4(N,ET1,ET2)
        END IF
  274   CONTINUE
```

```fortran
      DO 70 M= 1,NTN

          IF (NTN.EQ.15) THEN
            EA24T(ECL(NE,2),M,N)= EA24T(ECL(NE,2),M,N)+
     >         WRSJQ*PHI24(M,ET1,ET2,NE)*PHI24(N,ET1,ET2,NE)
          ELSE
            EA44T(ECL(NE,2),M,N)= EA44T(ECL(NE,2),M,N)+
     >         WRSJQ*PHI4(M,ET1,ET2)*PHI4(N,ET1,ET2)
          END IF

          SI= 0.
          IF (NTN.EQ.15) THEN
            DO 71 I= 1,2
   71          SI=SI+DPDX24(N,I,ET1,ET2,NE)*DPDX24(M,I,ET1,ET2,NE)
            EC24(ECL(NE,2),N,M)= EC24(ECL(NE,2),N,M)+ WRSJQ*SI
          ELSE
            DO 72 I= 1,2
   72          SI=SI+DPDX4(N,I,ET1,ET2,NE)*DPDX4(M,I,ET1,ET2,NE)
            EC44(ECL(NE,2),N,M)= EC44(ECL(NE,2),N,M)+ WRSJQ*SI
          END IF

          IF (NTN.EQ.15) THEN
            DO 73 I= 1,2
            DO 73 L= 1,NLNQ
   73          EB24(ECL(NE,2),N,L,M,I)= EB24(ECL(NE,2),N,L,M,I)+
     *            WRSJQ*PHI24(N,ET1,ET2,NE)*PHIQ(L,ET1,ET2)*
     *            DPDX24(M,I,ET1,ET2,NE)
          ELSE
            DO 74 I= 1,2
            DO 74 L= 1,NLNQ
   74          EB44(ECL(NE,2),N,L,M,I)= EB44(ECL(NE,2),N,L,M,I)+
     *            WRSJQ*PHI4(N,ET1,ET2)*PHIQ(L,ET1,ET2)*
     *            DPDX4(M,I,ET1,ET2,NE)
          END IF

   70 CONTINUE
      END IF
C
C Mixed quadratic and linear functions
C
      DO 41 N= 1,NLN
      DO 41 M= 1,NLNQ

      DO 41 I= 1,2

      IF (ECL(NE,1).EQ.0) THEN
        ED(ECL(NE,2),M,N,I)=ED(ECL(NE,2),M,N,I)+
     *     WRSJQ*PHIQ(M,ET1,ET2)*DPDXL(N,I,ET1,ET2,NE)
        EF(ECL(NE,2),N,M,I)=EF(ECL(NE,2),N,M,I)+
     *     WRSJQ*PHIL(N,ET1,ET2)*DPDXQ(M,I,ET1,ET2,NE)
      ELSE IF (ECL(NE,1).EQ.1) THEN
          IF (IPRESS.EQ.1) THEN
          EDLQ(ECL(NE,2),M,N,I)=EDLQ(ECL(NE,2),M,N,I)+
     *       WRSJQ*PHIQ(M,ET1,ET2)*DPDXLQ(N,I,ET1,ET2,NE)
          EFLQ(ECL(NE,2),N,M,I)=EFLQ(ECL(NE,2),N,M,I)+
     *       WRSJQ*PHILQ(N,ET1,ET2,NE)*DPDXQ(M,I,ET1,ET2,NE)
          ELSE
```

```fortran
      EDLQ(ECL(NE,2),M,N,I)=EDLQ(ECL(NE,2),M,N,I)+
*        WRSJQ*PHIQ(M,ET1,ET2)*DPDXL(N,I,ET1,ET2,NE)
      EFLQ(ECL(NE,2),N,M,I)=EFLQ(ECL(NE,2),N,M,I)+
*        WRSJQ*PHIL(N,ET1,ET2)*DPDXQ(M,I,ET1,ET2,NE)
      END IF
  ELSE
      IF (IPRESS.EQ.1) THEN
      EDQ(ECL(NE,2),M,N,I)=EDQ(ECL(NE,2),M,N,I)+
*        WRSJQ*PHIQ(M,ET1,ET2)*DPDXQ(N,I,ET1,ET2,NE)
      EFQ(ECL(NE,2),N,M,I)=EFQ(ECL(NE,2),N,M,I)+
*        WRSJQ*PHIQ(N,ET1,ET2)*DPDXQ(M,I,ET1,ET2,NE)
      ELSE
      EDQ(ECL(NE,2),M,N,I)=EDQ(ECL(NE,2),M,N,I)+
*        WRSJQ*PHIQ(M,ET1,ET2)*DPDXL(N,I,ET1,ET2,NE)
      EFQ(ECL(NE,2),N,M,I)=EFQ(ECL(NE,2),N,M,I)+
*        WRSJQ*PHIL(N,ET1,ET2)*DPDXQ(M,I,ET1,ET2,NE)
      END IF
  END IF

      DO 41 L= 1,NLNQ
      DO 41 J= 1,2

  IF (ECL(NE,1).EQ.0) THEN
    EG(ECL(NE,2),N,L,I,M,J)=EG(ECL(NE,2),N,L,I,M,J)+
*      WRSJQ*PHIL(N,ET1,ET2)*DPDXQ(L,I,ET1,ET2,NE)*
*      DPDXQ(M,J,ET1,ET2,NE)
    EH(ECL(NE,2),N,L,M,I,J)=EH(ECL(NE,2),N,L,M,I,J)+
*      WRSJQ*PHIL(N,ET1,ET2)*PHIQ(L,ET1,ET2)*
*      DPDXDYQ(M,I,J,R,S,NE)
  ELSE IF (ECL(NE,1).EQ.1) THEN
      IF (IPRESS.EQ.1) THEN
      EGLQ(ECL(NE,2),N,L,I,M,J)=EGLQ(ECL(NE,2),N,L,I,M,J)+
*        WRSJQ*PHILQ(N,ET1,ET2,NE)*DPDXQ(L,I,ET1,ET2,NE)*
*        DPDXQ(M,J,ET1,ET2,NE)
      EHLQ(ECL(NE,2),N,L,M,I,J)=EHLQ(ECL(NE,2),N,L,M,I,J)+
*        WRSJQ*PHILQ(N,ET1,ET2,NE)*PHIQ(L,ET1,ET2)*
*        DPDXDYQ(M,I,J,R,S,NE)
      ELSE
      EGLQ(ECL(NE,2),N,L,I,M,J)=EGLQ(ECL(NE,2),N,L,I,M,J)+
*        WRSJQ*PHIL(N,ET1,ET2)*DPDXQ(L,I,ET1,ET2,NE)*
*        DPDXQ(M,J,ET1,ET2,NE)
      EHLQ(ECL(NE,2),N,L,M,I,J)=EHLQ(ECL(NE,2),N,L,M,I,J)+
*        WRSJQ*PHIL(N,ET1,ET2)*PHIQ(L,ET1,ET2)*
*        DPDXDYQ(M,I,J,R,S,NE)
      END IF
  ELSE
      IF (IPRESS.EQ.1) THEN
      EGQ(ECL(NE,2),N,L,I,M,J)=EGQ(ECL(NE,2),N,L,I,M,J)+
*        WRSJQ*PHIQ(N,ET1,ET2)*DPDXQ(L,I,ET1,ET2,NE)*
*        DPDXQ(M,J,ET1,ET2,NE)
      EHQ(ECL(NE,2),N,L,M,I,J)=EHQ(ECL(NE,2),N,L,M,I,J)+
*        WRSJQ*PHIQ(N,ET1,ET2)*PHIQ(L,ET1,ET2)*
*        DPDXDYQ(M,I,J,R,S,NE)
      ELSE
      EGQ(ECL(NE,2),N,L,I,M,J)=EGQ(ECL(NE,2),N,L,I,M,J)+
*        WRSJQ*PHIL(N,ET1,ET2)*DPDXQ(L,I,ET1,ET2,NE)*
*        DPDXQ(M,J,ET1,ET2,NE)
      EHQ(ECL(NE,2),N,L,M,I,J)=EHQ(ECL(NE,2),N,L,M,I,J)+
```

```
*          WRSJQ*PHIL(N,ET1,ET2)*PHIQ(L,ET1,ET2)*
*          DPDXDYQ(M,I,J,R,S,NE)
         END IF
      END IF

   41 CONTINUE

      DO 30 N= 1,NLN
      DO 30 M= 1,NLN
C
         SI= 0.
         DO 39 I= 1,2
         IF (ECL(NE,1).EQ.0) THEN
           SI=SI+DPDXL(N,I,ET1,ET2,NE)*DPDXL(M,I,ET1,ET2,NE)
         ELSE IF (ECL(NE,1).EQ.1) THEN
             IF (IPRESS.EQ.1) THEN
             SI=SI+DPDXLQ(N,I,ET1,ET2,NE)*DPDXLQ(M,I,ET1,ET2,NE)
             ELSE
             SI=SI+DPDXL(N,I,ET1,ET2,NE)*DPDXL(M,I,ET1,ET2,NE)
             END IF
         ELSE IF (ECL(NE,1).EQ.2) THEN
             IF (IPRESS.EQ.1) THEN
             SI=SI+DPDXQ(N,I,ET1,ET2,NE)*DPDXQ(M,I,ET1,ET2,NE)
             ELSE
             SI=SI+DPDXL(N,I,ET1,ET2,NE)*DPDXL(M,I,ET1,ET2,NE)
             END IF
         END IF
   39    CONTINUE
         IF (ECL(NE,1).EQ.0) THEN
           EE(ECL(NE,2),N,M)=EE(ECL(NE,2),N,M)+WRSJQ*SI
         ELSE IF (ECL(NE,1).EQ.1) THEN
           EELQ(ECL(NE,2),N,M)=EELQ(ECL(NE,2),N,M)+WRSJQ*SI
         ELSE IF (ECL(NE,1).EQ.2) THEN
           EEQ(ECL(NE,2),N,M)=EEQ(ECL(NE,2),N,M)+WRSJQ*SI
         END IF
C
   30 CONTINUE

C Pressure boundary tensor formulation

      NE= 0
      DO 50 NES= 1,NESURF(4)
      NE= NE+1

C Coordinate coefficients

      DO 51 N= 1,2

C Quadratic interpolation function (corners and midnodes)

         A2(N)=   4.*XG(CQ(NE,9),N)
         B2(N)=   2.*XG(CQ(NE,4),N)-2.*XG(CQ(NE,8),N)
         C2(N)= -2.*XG(CQ(NE,2),N)+2.*XG(CQ(NE,6),N)
         D2(N)=      XG(CQ(NE,1),N)-    XG(CQ(NE,3),N)
     *             +XG(CQ(NE,5),N)-    XG(CQ(NE,7),N)
         E2(N)=   2.*XG(CQ(NE,4),N)+2.*XG(CQ(NE,8),N)
     *           -4.*XG(CQ(NE,9),N)
         F2(N)=   2.*XG(CQ(NE,2),N)+2.*XG(CQ(NE,6),N)
```

```fortran
      *                -4.*XG(CQ(NE,9),N)
          G2(N)= -1.*XG(CQ(NE,1),N)+    XG(CQ(NE,3),N)
      *             +XG(CQ(NE,5),N)-    XG(CQ(NE,7),N)
      *            -2.*XG(CQ(NE,4),N)+2.*XG(CQ(NE,8),N)
          H2(N)= -1.*XG(CQ(NE,1),N)-    XG(CQ(NE,3),N)
      *             +XG(CQ(NE,5),N)+    XG(CQ(NE,7),N)
      *            +2.*XG(CQ(NE,2),N)-2.*XG(CQ(NE,6),N)
          P2(N)=     XG(CQ(NE,1),N)+    XG(CQ(NE,3),N)
      *             +XG(CQ(NE,5),N)+    XG(CQ(NE,7),N)
      *            -2.*XG(CQ(NE,2),N)-2.*XG(CQ(NE,4),N)
      *            -2.*XG(CQ(NE,6),N)-2.*XG(CQ(NE,8),N)
      *             +4.*XG(CQ(NE,9),N)

   51 CONTINUE

      DO 50 S= 1,NPOINT

      ET1= XI(S,1)
      ET2= 1.

C Coefficients used in 2nd order Jacobian and derivative calculations

      DO 53 P= 1,2
        A(P)= B2(P)+D2(P)*ET2+2.*E2(P)*ET1+G2(P)*ET2**2
      *        +2.*H2(P)*ET1*ET2 +2.*P2(P)*ET1*ET2**2
        B(P)= C2(P)+D2(P)*ET1+2.*F2(P)*ET2+H2(P)*ET1**2
      *        +2.*G2(P)*ET1*ET2 +2.*P2(P)*ET2*ET1**2
        AD1(P)= 2.*(E2(P)+H2(P)*ET2+P2(P)*ET2**2)
        AD2(P)= D2(P)+2.*G2(P)*ET2+2.*H2(P)*ET1+4.*P2(P)*ET1*ET2
        BD1(P)= AD2(P)
        BD2(P)= 2.*(F2(P)+G2(P)*ET1+P2(P)*ET1**2)
   53 CONTINUE

C Jacobian determinant: quadratic variation

      DETJQ= 0.
      DO 54 N= 1,2
      DO 54 M= 1,2
        IF (M.EQ.N) GOTO 54
        DETJQ= EPS(M,N)/16.*A(M)*B(N)+DETJQ
   54 CONTINUE

C Inverse Jacobian tensor

      DO 55 I= 1,2
      DO 55 K= 1,2
        IF (I.EQ.1) THEN
        P= 2
        SIGN= 1.
        ELSE
        P= 1
        SIGN= -1.
        END IF
        IF (K.EQ.1) THEN
          JINV(I,K)= SIGN/4./DETJQ*B(P)
        ELSE
          JINV(I,K)= -1.*SIGN/4./DETJQ*A(P)
        END IF
```

212

```fortran
   55 CONTINUE

         IF (NE.EQ.1.OR.NE.EQ.NESURF(4)) THEN
         X= XI(S,1)
         DO 59 N= 1,5
           IF (N.EQ.1) THEN
             FAC= 2.*(X**4-X**3-.25*X**2+.25*X)
           ELSE IF (N.EQ.2) THEN
             FAC=-8.*(X**4-.5*X**3-X**2+.5*X)
           ELSE IF (N.EQ.3) THEN
             FAC=12.*(X**4-1.25*X**2+.25)
           ELSE IF (N.EQ.4) THEN
             FAC=-8.*(X**4+.5*X**3-X**2-.5*X)
           ELSE IF (N.EQ.5) THEN
             FAC= 2.*(X**4+X**3-.25*X**2-.25*X)
           END IF
           DO 59 M= 1,25
             SAQ2(ECL(NE,2),N,M)= SAQ2(ECL(NE,2),N,M)+
     >         W(S)*FAC*DPDX4(M,1,ET1,ET2,NE)/6.
   59     CONTINUE
         END IF

         DO 50 N= 1,3
           IF (N.EQ.1) FAC= XI(S,1)*(XI(S,1)-1.)
           IF (N.EQ.2) FAC= -2.*(XI(S,1)**2-1.)
           IF (N.EQ.3) FAC= XI(S,1)*(XI(S,1)+1.)
           DO 50 M= 1,NLNQ
             SAQ(NES,N,M)= SAQ(NES,N,M)+
     >         W(S)*FAC*DPDXQ(M,1,ET1,ET2,NE)/4.
   50 CONTINUE

         RETURN
         END
```

H.   Function EPS

```fortran
         FUNCTION EPS(M,N)

C***********************************************************************
C 2-index permutation symbol. Outputs
C = 1 if m=1 and n=2
C =-1 if m=2 and n=1
C = 0 if m=n
C***********************************************************************

         IF (M.EQ.1.AND.N.EQ.2) THEN
           A= 1.
         ELSE IF (M.EQ.2.AND.N.EQ.1) THEN
           A= -1.
         ELSE
           A= 0.
         END IF
C
         EPS= A
C
         RETURN
```

```
            END



I.      Function DPDXL

        FUNCTION DPDXL(N,I,ET1,ET2,NE)

C*******************************************************************
C Calculates first order derivative of first order curved isoparameteric
C*******************************************************************

        COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

        COMMON/VQUAD/
       * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
       * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

        REAL JINV

        EN1= XINL(N,1)
        EN2= XINL(N,2)

        CN= .25
        CN1= EN1+EN1*EN2*ET2
        CN2= EN2+EN1*EN2*ET1

        DPDXL= CN*(CN1*JINV(I,1)+CN2*JINV(I,2))

        RETURN
        END



J.      Function DPDXLQ

        FUNCTION DPDXLQ(N,I,ET1,ET2,NE)

C*******************************************************************
C Calculates first order derivative of second order isoparameteric function
C*******************************************************************

        COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

        COMMON/VQUAD/
       * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
       * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

        COMMON/SURFTYPE/ NESURF(4)

        REAL JINV

        IF (NE.LE.NESURF(4)) THEN
        EN1= XINLQ(N,1,2)
        EN2= XINLQ(N,2,2)
        IF (N.EQ.3.OR.N.EQ.6) THEN
          CN= .5
          CN1= EN1*(1.-ET2**2)
```

```
          CN2= -2.*ET2*(EN1*ET1+1.)
       ELSE
          CN= .25
          CN1= EN1*(ET2**2+ET2*EN2)
          CN2= (2.*ET2+EN2)*(EN1*ET1+1.)
       END IF
       ELSE
       EN1= XINLQ(N,1,1)
       EN2= XINLQ(N,2,1)
       IF (N.EQ.2.OR.N.EQ.5) THEN
          CN= .5
          CN1= -2.*ET1*(EN2*ET2+1.)
          CN2= EN2*(1.-ET1**2)
       ELSE
          CN= .25
          CN1= (2.*ET1+EN1)*(EN2*ET2+1.)
          CN2= EN2*(ET1**2+ET1*EN1)
       END IF
       END IF

       DPDXLQ= CN*(CN1*JINV(I,1)+CN2*JINV(I,2))

       RETURN
       END
```

K.   Function DPDXQ

```
       FUNCTION DPDXQ(N,I,ET1,ET2,NE)

C*********************************************************************
C Calculates first order derivative of second order isoparameteric
C function
C*********************************************************************

       COMMON/VQUAD/
     * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

       REAL JINV

       EN1= XINQ(N,1)
       EN2= XINQ(N,2)

       IF (N.EQ.9) THEN
          CN= -2.
          CN1= ET1*(1.-ET2**2)
       CN2= ET2*(1.-ET1**2)
       ELSE IF (N.EQ.2.OR.N.EQ.6) THEN
          CN= EN2/2.
       CN1= -2.*ET1*ET2*(EN2*ET2+1.)
       CN2= (1.-ET1**2)*(2.*EN2*ET2+1.)
       ELSE IF (N.EQ.4.OR.N.EQ.8) THEN
          CN= EN1/2.
          CN1= (1.-ET2**2)*(2.*EN1*ET1+1.)
       CN2= -2.*ET1*ET2*(EN1*ET1+1.)
       ELSE
```

```
         CN= EN1*EN2/4.
        CN1= ET2*(1.+EN2*ET2)*(2.*EN1*ET1+1.)
        CN2= ET1*(1.+EN1*ET1)*(2.*EN2*ET2+1.)
        END IF
C
        DPDXQ= CN*(CN1*JINV(I,1)+CN2*JINV(I,2))
C
        RETURN
        END
```

L.   Function DPDX24

```
        FUNCTION DPDX24(N,I,ET1,ET2,NE)

C*********************************************************************
C Derivative of mixed 2nd/4th-order quadrilateral element
C*********************************************************************

        COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

        COMMON/VQUAD/
      * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
      * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

        COMMON/SURFTYPE/ NESURF(4)

        REAL JINV

        IF (NE.LE.NESURF(4)) THEN

        IF (N.EQ.1) THEN
        IGROUP= 1
        C=   .083333333333       ! =  1/12
        EN1= -1.
        EN2= -1.
        ELSE IF (N.EQ.2) THEN
        IGROUP= 2
        C=   .1666666666667      ! =  1/6
        EN1=  0.
        EN2= -1.
        ELSE IF (N.EQ.3) THEN
        IGROUP= 1
        C=   .083333333333       ! =  1/12
        EN1=  1.
        EN2= -1.
        ELSE IF (N.EQ.4) THEN
        IGROUP= 1
        C=-1.333333333333        ! = -8/6
        EN1=  1.
        EN2= -.5
        ELSE IF (N.EQ.5) THEN
        IGROUP= 3
        C=  2.                   ! =  2
        EN1=  1.
        EN2=  0.
        ELSE IF (N.EQ.6) THEN
```

216

```
IGROUP= 1
C=-1.333333333333      ! = -8/6
EN1=  1.
EN2=  .5
ELSE IF (N.EQ.7) THEN
IGROUP= 1
C=  .083333333333      ! =  1/12
EN1=  1.
EN2=  1.
ELSE IF (N.EQ.8) THEN
IGROUP= 2
C=  .16666666666667    ! =  1/6
EN1=  0.
EN2=  1.
ELSE IF (N.EQ.9) THEN
IGROUP= 1
C=  .083333333333      ! =  1/12
EN1= -1.
EN2=  1.
ELSE IF (N.EQ.10) THEN
IGROUP= 1
C=-1.333333333333      ! = -8/6
EN1= -1.
EN2=  .5
ELSE IF (N.EQ.11) THEN
IGROUP= 3
C=  2.                 ! =  2
EN1= -1.
EN2=  0.
ELSE IF (N.EQ.12) THEN
IGROUP= 1
C=-1.333333333333      ! = -8/6
EN1= -1.
EN2= -.5
ELSE IF (N.EQ.13) THEN
IGROUP= 2
C=-2.6666666666667     ! =  -8/3
EN1=  0.
EN2= -.5
ELSE IF (N.EQ.14) THEN
IGROUP= 4
C=  4.                 ! =  4
EN1=  0.
EN2=  0.
ELSE IF (N.EQ.15) THEN
IGROUP= 2
C=-2.6666666666667     ! =  -8/3
EN1=  0.
EN2=  .5
END IF

ELSE

IF (N.EQ.1) THEN
IGROUP= 1
C=  .08333333333333    ! =  1/12
EN1= -1.
EN2= -1.
```

```fortran
      ELSE IF (N.EQ.2) THEN
      IGROUP= 1
      C=-1.33333333333333     ! = -8/6
      EN1= -.5
      EN2= -1.
      ELSE IF (N.EQ.3) THEN
      IGROUP= 3
      C= 2.                   ! =  2
      EN1=  0.
      EN2= -1.
      ELSE IF (N.EQ.4) THEN
      IGROUP= 1
      C=-1.33333333333333     ! = -8/6
      EN1=  .5
      EN2= -1.
      ELSE IF (N.EQ.5) THEN
      IGROUP= 1
      C=  .08333333333333     ! =  1/12
      EN1=  1.
      EN2= -1.
      ELSE IF (N.EQ.6) THEN
      IGROUP= 2
      C=  .16666666666667     ! = -8/6
      EN1=  1.
      EN2=  0.
      ELSE IF (N.EQ.7) THEN
      IGROUP= 1
      C=  .08333333333333     ! =  1/12
      EN1=  1.
      EN2=  1.
      ELSE IF (N.EQ.8) THEN
      IGROUP= 1
      C=-1.33333333333333     ! = -8/6
      EN1=  .5
      EN2=  1.
      ELSE IF (N.EQ.9) THEN
      IGROUP= 3
      C= 2.                   ! =  2
      EN1=  0.
      EN2=  1.
      ELSE IF (N.EQ.10) THEN
      IGROUP= 1
      C=-1.33333333333333     ! = -8/6
      EN1= -.5
      EN2=  1.
      ELSE IF (N.EQ.11) THEN
      IGROUP= 1
      C=  .08333333333333     ! =  1/12
      EN1= -1.
      EN2=  1.
      ELSE IF (N.EQ.12) THEN
      IGROUP= 2
      C=  .16666666666667     ! = 1/6
      EN1= -1.
      EN2=  0.
      ELSE IF (N.EQ.13) THEN
      IGROUP= 2
      C=-2.6666666666667      ! =  -8/3
```

```
      EN1= -.5
      EN2=  0.
      ELSE IF (N.EQ.14) THEN
      IGROUP= 4
      C= 4.                        ! = 4
      EN1=  0.
      EN2=  0.
      ELSE IF (N.EQ.15) THEN
      IGROUP= 2
      C=-2.6666666666667    ! =  -8/3
      EN1=  .5
      EN2=  0.
      END IF

      END IF

      IF (NE.LE.NESURF(4)) THEN
      ENA= EN2
      ENB= EN1
      ETA= ET2
      ETB= ET1
      CJA= JINV(I,2)
      CJB= JINV(I,1)
      ELSE
      ENA= EN1
      ENB= EN2
      ETA= ET1
      ETB= ET2
      CJA= JINV(I,1)
      CJB= JINV(I,2)
      END IF

      IF (IGROUP.EQ.1) THEN
      CNA= (16.*ENA**2*ETA**3+ 12.*ENA**3*ETA**2- 2.*ETA- ENA)*
     >      (ETB**2+ ENB*ETB)
      CNB= (4.*ENA**2*ETA**4+ 4.*ENA**3*ETA**3- ETA**2- ENA*ETA)*
     >      (2.*ETB+ ENB)
      ELSE IF (IGROUP.EQ.2) THEN
      CNA= (16.*ENA**2*ETA**3+ 12.*ENA**3*ETA**2- 2.*ETA- ENA)*
     >      (1.- ETB**2)
      CNB= (4.*ENA**2*ETA**4+ 4.*ENA**3*ETA**3- ETA**2- ENA*ETA)*
     >      (-2.*ETB)
      ELSE IF (IGROUP.EQ.3) THEN
      CNA= (4.*ETA**3- 2.5*ETA)*
     >      (ETB**2+ ENB*ETB)
      CNB= (ETA**4- 1.25*ETA**2+ .25)*
     >      (2.*ETB+ ENB)
      ELSE IF (IGROUP.EQ.4) THEN
      CNA= (4.*ETA**3- 2.5*ETA)*(1.- ETB**2)
      CNB= (ETA**4- 1.25*ETA**2+ .25)*(-2.*ETB)
      END IF
C
      DPDX24= C*(CNA*CJA+CNB*CJB)
C
      RETURN
      END
      END
```

M.　Function DPDX4

```
      FUNCTION DPDX4(N,I,ET1,ET2,NE)

C**********************************************************************
C Calculates first order derivative of fourth order isoparameteric
C function
C**********************************************************************

      COMMON/VQUAD/
     * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

      REAL JINV

      IF (N.EQ.1) THEN
      IGROUP= 1
      C=   .027777777778      ! =  1/36
      EN1= -1.
      EN2= -1.
      ELSE IF (N.EQ.2) THEN
      IGROUP= 1
      C= -.444444444444       ! = -8/18
      EN1= -.5
      EN2= -1.
      ELSE IF (N.EQ.3) THEN
      IGROUP= 2
      C=   .6666666666667      ! =  2/3
      EN1=  0.
      EN2= -1.
      ELSE IF (N.EQ.4) THEN
      IGROUP= 1
      C= -.444444444444       ! = -8/18
      EN1=  .5
      EN2= -1.
      ELSE IF (N.EQ.5) THEN
      IGROUP= 1
      C=   .027777777778      ! =  1/36
      EN1=  1.
      EN2= -1.
      ELSE IF (N.EQ.6) THEN
      IGROUP= 1
      C= -.444444444444       ! = -8/18
      EN1=  1.
      EN2= -.5
      ELSE IF (N.EQ.7) THEN
      IGROUP= 3
      C=   .6666666666667      ! =  2/3
      EN1=  1.
      EN2=  0.
      ELSE IF (N.EQ.8) THEN
      IGROUP= 1
      C= -.444444444444       ! = -8/18
      EN1=  1.
      EN2=  .5
      ELSE IF (N.EQ.9) THEN
```

```
IGROUP= 1
C=  .027777777778      ! =  1/36
EN1=  1.
EN2=  1.
ELSE IF (N.EQ.10) THEN
IGROUP= 1
C= -.444444444444      ! = -8/18
EN1=  .5
EN2=   1.
ELSE IF (N.EQ.11) THEN
IGROUP= 2
C=  .6666666666667     ! =  2/3
EN1=  0.
EN2=  1.
ELSE IF (N.EQ.12) THEN
IGROUP= 1
C= -.444444444444      ! = -8/18
EN1= -.5
EN2=   1.
ELSE IF (N.EQ.13) THEN
IGROUP= 1
C=  .027777777778      ! =  1/36
EN1= -1.
EN2=  1.
ELSE IF (N.EQ.14) THEN
IGROUP= 1
C= -.444444444444      ! = -8/18
EN1= -1.
EN2=   .5
ELSE IF (N.EQ.15) THEN
IGROUP= 3
C=  .6666666666667     ! =  2/3
EN1= -1.
EN2=   0.
ELSE IF (N.EQ.16) THEN
IGROUP= 1
C= -.444444444444      ! = -8/18
EN1= -1.
EN2= -.5
ELSE IF (N.EQ.17) THEN
IGROUP= 1
C= 7.111111111111      ! =  64/9
EN1= -.5
EN2= -.5
ELSE IF (N.EQ.18) THEN
IGROUP= 2
C=-10.666666666667     ! = -32/3
EN1=  0.
EN2= -.5
ELSE IF (N.EQ.19) THEN
IGROUP= 1
C= 7.111111111111      ! =  64/9
EN1=  .5
EN2= -.5
ELSE IF (N.EQ.20) THEN
IGROUP= 3
C=-10.666666666667     ! = -32/3
EN1=  .5
```

```
      EN2=  0.
      ELSE IF (N.EQ.21) THEN
      IGROUP= 1
      C= 7.111111111111       ! =  64/9
      EN1=  .5
      EN2=  .5
      ELSE IF (N.EQ.22) THEN
      IGROUP= 2
      C=-10.666666666667       ! = -32/3
      EN1=  0.
      EN2=  .5
      ELSE IF (N.EQ.23) THEN
      IGROUP= 1
      C= 7.111111111111       ! =  64/9
      EN1= -.5
      EN2=  .5
      ELSE IF (N.EQ.24) THEN
      IGROUP= 3
      C=-10.666666666667       ! = -32/3
      EN1= -.5
      EN2=  0.
      ELSE IF (N.EQ.25) THEN
      IGROUP= 4
      C= 16.
      EN1=  0.
      EN2=  0.
      END IF

      IF (IGROUP.EQ.1) THEN
      CN1= (16.*EN1**2*ET1**3+ 12.*EN1**3*ET1**2- 2.*ET1- EN1)*
     >       (4.*EN2**2*ET2**4+ 4.*EN2**3*ET2**3- ET2**2- EN2*ET2)
      CN2= (4.*EN1**2*ET1**4+ 4.*EN1**3*ET1**3- ET1**2- EN1*ET1)*
     >       (16.*EN2**2*ET2**3+ 12.*EN2**3*ET2**2- 2.*ET2- EN2)
      ELSE IF (IGROUP.EQ.2) THEN
      CN1= (4.*ET1**3- 2.5*ET1)*
     >       (4.*EN2**2*ET2**4+ 4.*EN2**3*ET2**3- ET2**2- EN2*ET2)
      CN2= (ET1**4- 1.25*ET1**2+ .25)*
     >       (16.*EN2**2*ET2**3+ 12.*EN2**3*ET2**2- 2.*ET2- EN2)
      ELSE IF (IGROUP.EQ.3) THEN
      CN1= (16.*EN1**2*ET1**3+ 12.*EN1**3*ET1**2- 2.*ET1- EN1)*
     >       (ET2**4- 1.25*ET2**2+ .25)
      CN2= (4.*EN1**2*ET1**4+ 4.*EN1**3*ET1**3- ET1**2- EN1*ET1)*
     >       (4.*ET2**3- 2.5*ET2)
      ELSE IF (IGROUP.EQ.4) THEN
      CN1= (4.*ET1**3- 2.5*ET1)*
     >       (ET2**4- 1.25*ET2**2+ .25)
      CN2= (ET1**4- 1.25*ET1**2+ .25)*
     >       (4.*ET2**3- 2.5*ET2)
      END IF

      DPDX4= C*(CN1*JINV(I,1)+CN2*JINV(I,2))
C
      RETURN
      END
```

N.   Function DPDXDYQ

```fortran
      FUNCTION DPDXDYQ(N,I,J,NXI,NETA,NE)

C*********************************************************************
C Calculates second order derivative of second order isoparameteric
C function
C*********************************************************************
C
      COMMON/VINTEG/XI(10,2),W(10)
C
      COMMON/VQUAD/
     * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)
C
      INTEGER P
      REAL JINV
C
      EN1= XINQ(N,1)
      EN2= XINQ(N,2)
      ET1= XI(NXI,1)
      ET2= XI(NETA,2)
C
      IF (I.EQ.1) THEN
        EIP= 1.
      P= 2
      ELSE
        EIP= -1.
      P= 1
      END IF
C
      IF (N.EQ.9) THEN
        CN= -2.
        CN1= ET1*(1.-ET2**2)
      CN2= ET2*(1.-ET1**2)
      DN1= 1.-ET2**2
      DN2= -2.*ET1*ET2
      DN3= DN2
      DN4= 1.-ET1**2
      ELSE IF (N.EQ.2.OR.N.EQ.6) THEN
        CN= EN2/2.
      CN1= -2.*ET1*ET2*(EN2*ET2+1.)
      CN2= (1.-ET1**2)*(2.*EN2*ET2+1.)
      DN1= -2.*ET2*(EN2*ET2+1.)
      DN2= -2.*ET1*(2.*EN2*ET2+1.)
      DN3= DN2
      DN4= 2.*EN2*(1.-ET1**2)
      ELSE IF (N.EQ.4.OR.N.EQ.8) THEN
        CN= EN1/2.
        CN1= (1.-ET2**2)*(2.*EN1*ET1+1.)
      CN2= -2.*ET1*ET2*(EN1*ET1+1.)
      DN1= 2.*EN1*(1.-ET2**2)
      DN2= -2.*ET2*(2.*EN1*ET1+1.)
      DN3= DN2
      DN4= -2.*ET1*(EN1*ET1+1.)
      ELSE
        CN= EN1*EN2/4.
      CN1= ET2*(1.+EN2*ET2)*(2.*EN1*ET1+1.)
      CN2= ET1*(1.+EN1*ET1)*(2.*EN2*ET2+1.)
      DN1= 2.*EN1*ET2*(EN2*ET2+1.)
```

```
      DN2= (2.*EN1*ET1+1.)*(2.*EN2*ET2+1.)
      DN3= DN2
      DN4= 2.*EN2*ET1*(EN1*ET1+1.)
      END IF
C
      DPDXDYQ= EIP*CN/4./DETJQ*
     * ((CN1*BD1(P)-CN2*AD1(P)+B(P)*DN1-A(P)*DN3)*JINV(J,1)+
     *  (CN1*BD2(P)-CN2*AD2(P)+B(P)*DN2-A(P)*DN4)*JINV(J,2))
C
      RETURN
      END
```

O.   Function PHIL

```
      FUNCTION PHIL(N,ET1,ET2)

C************************************************************************
C Linear isoparametric function for quadrilateral element
C************************************************************************

      COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

      PHIL= (1.+XINL(N,1)*ET1)*(1.+XINL(N,2)*ET2)/4.

      RETURN
      END
```

P.   Function PHILQ

```
      FUNCTION PHILQ(N,ET1,ET2,NE)

C************************************************************************
C Second-order isoparametric function for quadrilateral element
C************************************************************************

      COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

      COMMON/VQUAD/
     * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

      COMMON/SURFTYPE/ NESURF(4)

      REAL JINV

      IF (NE.LE.NESURF(4)) THEN
      EN1= XINLQ(N,1,2)
      EN2= XINLQ(N,2,2)
      IF (N.EQ.3.OR.N.EQ.6) THEN
         X= .5*(1.-ET2**2)*(1.+EN1*ET1)
      ELSE
         X= .25*ET2*(ET2+EN2)*(EN1*ET1+1.)
      END IF
      ELSE
```

224

```
      EN1= XINLQ(N,1,1)
      EN2= XINLQ(N,2,1)
      IF (N.EQ.2.OR.N.EQ.5) THEN
        X= .5*(1.-ET1**2)*(1.+EN2*ET2)
      ELSE
        X= .25*ET1*(ET1+EN1)*(EN2*ET2+1.)
      END IF
      END IF

      PHILQ= X

      RETURN
      END
```

Q. Function PHIQ

```
      FUNCTION PHIQ(N,ET1,ET2)

C**********************************************************************
C FUNCTION PHIQ
C Second-order isoparametric function for quadrilateral element
C**********************************************************************

      COMMON/VQUAD/
     * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)
C
      REAL JINV
C
      EN1= XINQ(N,1)
      EN2= XINQ(N,2)
C
      IF (N.EQ.9) THEN
        X= (1.-ET1**2)*(1.-ET2**2)
      ELSE IF (N.EQ.2.OR.N.EQ.6) THEN
        X= .5*(1.-ET1**2)*EN2*ET2*(1.+EN2*ET2)
      ELSE IF (N.EQ.4.OR.N.EQ.8) THEN
        X= .5*(1.-ET2**2)*EN1*ET1*(1.+EN1*ET1)
      ELSE
        X= .25*(1.+EN1*ET1)*(1.+EN2*ET2)*EN1*ET1*EN2*ET2
      END IF
C
      PHIQ= X
C
      RETURN
      END
```

R. Function PHI24

```
      FUNCTION PHI24(N,ET1,ET2,NE)

C**********************************************************************
C Fourth-order isoparametric function for quadrilateral element
C**********************************************************************
```

```
      COMMON/VLIN/A1(2),B1(2),C1(2),D1(2),XINL(4,2),XINLQ(6,2,2),DETJL

      COMMON/VQUAD/
     * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)

      COMMON/SURFTYPE/ NESURF(4)

      REAL JINV

      IF (NE.LE.NESURF(4)) THEN

      IF (N.EQ.1) THEN
      IGROUP= 1
      C=  .083333333333      ! =  1/12
      EN1= -1.
      EN2= -1.
      ELSE IF (N.EQ.2) THEN
      IGROUP= 2
      C=  .1666666666667     ! =  1/6
      EN1=  0.
      EN2= -1.
      ELSE IF (N.EQ.3) THEN
      IGROUP= 1
      C=  .083333333333      ! =  1/12
      EN1=  1.
      EN2= -1.
      ELSE IF (N.EQ.4) THEN
      IGROUP= 1
      C=-1.333333333333      ! = -8/6
      EN1=  1.
      EN2= -.5
      ELSE IF (N.EQ.5) THEN
      IGROUP= 3
      C=  2.                 ! =  2
      EN1=  1.
      EN2=  0.
      ELSE IF (N.EQ.6) THEN
      IGROUP= 1
      C=-1.333333333333      ! = -8/6
      EN1=  1.
      EN2=  .5
      ELSE IF (N.EQ.7) THEN
      IGROUP= 1
      C=  .083333333333      ! =  1/12
      EN1=  1.
      EN2=  1.
      ELSE IF (N.EQ.8) THEN
      IGROUP= 2
      C=  .1666666666667     ! =  1/6
      EN1=  0.
      EN2=  1.
      ELSE IF (N.EQ.9) THEN
      IGROUP= 1
      C=  .083333333333      ! =  1/12
      EN1= -1.
      EN2=  1.
```

```
      ELSE IF (N.EQ.10) THEN
      IGROUP= 1
      C=-1.333333333333       ! = -8/6
      EN1= -1.
      EN2=  .5
      ELSE IF (N.EQ.11) THEN
      IGROUP= 3
      C=  2.                   ! =  2
      EN1= -1.
      EN2=  0.
      ELSE IF (N.EQ.12) THEN
      IGROUP= 1
      C=-1.333333333333       ! = -8/6
      EN1= -1.
      EN2= -.5
      ELSE IF (N.EQ.13) THEN
      IGROUP= 2
      C=-2.6666666666667      ! =  -8/3
      EN1=  0.
      EN2= -.5
      ELSE IF (N.EQ.14) THEN
      IGROUP= 4
      C= 4.                    ! = 4
      EN1=  0.
      EN2=  0.
      ELSE IF (N.EQ.15) THEN
      IGROUP= 2
      C=-2.6666666666667      ! =  -8/3
      EN1=  0.
      EN2=  .5
      END IF

      ELSE

      IF (N.EQ.1) THEN
      IGROUP= 1
      C=  .08333333333333     ! =  1/12
      EN1= -1.
      EN2= -1.
      ELSE IF (N.EQ.2) THEN
      IGROUP= 1
      C=-1.33333333333333     ! = -8/6
      EN1= -.5
      EN2= -1.
      ELSE IF (N.EQ.3) THEN
      IGROUP= 3
      C= 2.                    ! =  2
      EN1=  0.
      EN2= -1.
      ELSE IF (N.EQ.4) THEN
      IGROUP= 1
      C=-1.33333333333333     ! = -8/6
      EN1=  .5
      EN2= -1.
      ELSE IF (N.EQ.5) THEN
      IGROUP= 1
      C=  .08333333333333     ! =  1/12
      EN1=  1.
```

```
      EN2= -1.
      ELSE IF (N.EQ.6) THEN
      IGROUP= 2
      C=  .16666666666667      ! =   1/6
      EN1=  1.
      EN2=  0.
      ELSE IF (N.EQ.7) THEN
      IGROUP= 1
      C=  .08333333333333      ! =   1/12
      EN1=  1.
      EN2=  1.
      ELSE IF (N.EQ.8) THEN
      IGROUP= 1
      C=-1.33333333333333      ! =  -8/6
      EN1=  .5
      EN2=  1.
      ELSE IF (N.EQ.9) THEN
      IGROUP= 3
      C= 2.                    ! =   2
      EN1=  0.
      EN2=  1.
      ELSE IF (N.EQ.10) THEN
      IGROUP= 1
      C=-1.33333333333333      ! =  -8/6
      EN1= -.5
      EN2=  1.
      ELSE IF (N.EQ.11) THEN
      IGROUP= 1
      C=  .08333333333333      ! =   1/12
      EN1= -1.
      EN2=  1.
      ELSE IF (N.EQ.12) THEN
      IGROUP= 2
      C=  .16666666666667      ! =  1/6
      EN1= -1.
      EN2=  0.
      ELSE IF (N.EQ.13) THEN
      IGROUP= 2
      C=-2.6666666666667       ! =  -8/3
      EN1= -.5
      EN2=  0.
      ELSE IF (N.EQ.14) THEN
      IGROUP= 4
      C= 4.                    ! =  4
      EN1=  0.
      EN2=  0.
      ELSE IF (N.EQ.15) THEN
      IGROUP= 2
      C=-2.6666666666667       ! =  -8/3
      EN1=  .5
      EN2=  0.
      END IF

      END IF

      IF (NE.LE.NESURF(4)) THEN
      ENA= EN2
      ENB= EN1
```

```
      ETA= ET2
      ETB= ET1
      ELSE
      ENA= EN1
      ENB= EN2
      ETA= ET1
      ETB= ET2
      END IF

      IF (IGROUP.EQ.1) THEN
      X= C*(4.*ENA**2*ETA**4+ 4.*ENA**3*ETA**3- ETA**2- ENA*ETA)*
     >       (ETB**2+ ENB*ETB)
      ELSE IF (IGROUP.EQ.2) THEN
      X= C*(4.*ENA**2*ETA**4+ 4.*ENA**3*ETA**3- ETA**2- ENA*ETA)*
     >       (1.- ETB**2)
      ELSE IF (IGROUP.EQ.3) THEN
      X= C*(ETA**4- 1.25*ETA**2+ .25)*
     >       (ETB**2+ ENB*ETB)
      ELSE IF (IGROUP.EQ.4) THEN
      X= C*(ETA**4- 1.25*ETA**2+ .25)*(1.- ETB**2)
      END IF
C
      PHI24= X
C
      RETURN
      END
```

S.   Function PHI4

```
      FUNCTION PHI4(N,ET1,ET2)

C*********************************************************************
C Fourth-order isoparametric function for quadrilateral element
C*********************************************************************

      COMMON/VQUAD/
     * A2(2),B2(2),C2(2),D2(2),E2(2),F2(2),G2(2),H2(2),P2(2),
     * XINQ(9,2),DETJQ,JINV(2,2),A(2),B(2),AD1(2),AD2(2),BD1(2),BD2(2)
      REAL JINV
C
      IF (N.EQ.1) THEN
        IGROUP= 1
        C=  .027777777778     ! =  1/36
        EN1= -1.
        EN2= -1.
      ELSE IF (N.EQ.2) THEN
        IGROUP= 1
        C= -.444444444444     ! = -8/18
        EN1= -.5
        EN2= -1.
        ELSE IF (N.EQ.3) THEN
      IGROUP= 2
        C=  .6666666666667     ! =  2/3
        EN1=  0.
        EN2= -1.
      ELSE IF (N.EQ.4) THEN
```

```
      IGROUP= 1
      C= -.444444444444      ! = -8/18
      EN1=  .5
      EN2= -1.
   ELSE IF (N.EQ.5) THEN
      IGROUP= 1
      C=  .027777777778       ! =  1/36
      EN1=  1.
      EN2= -1.
   ELSE IF (N.EQ.6) THEN
      IGROUP= 1
      C= -.444444444444      ! = -8/18
      EN1=  1.
      EN2= -.5
   ELSE IF (N.EQ.7) THEN
      IGROUP= 3
      C=  .6666666666667      ! =  2/3
      EN1=  1.
      EN2=  0.
   ELSE IF (N.EQ.8) THEN
      IGROUP= 1
      C= -.444444444444      ! = -8/18
      EN1=  1.
      EN2=  .5
   ELSE IF (N.EQ.9) THEN
      IGROUP= 1
      C=  .027777777778       ! =  1/36
      EN1=  1.
      EN2=  1.
   ELSE IF (N.EQ.10) THEN
      IGROUP= 1
      C= -.444444444444      ! = -8/18
      EN1=  .5
      EN2=  1.
   ELSE IF (N.EQ.11) THEN
      IGROUP= 2
      C=  .6666666666667      ! =  2/3
      EN1=  0.
      EN2=  1.
   ELSE IF (N.EQ.12) THEN
      IGROUP= 1
      C= -.444444444444      ! = -8/18
      EN1= -.5
      EN2=  1.
   ELSE IF (N.EQ.13) THEN
      IGROUP= 1
      C=  .027777777778       ! =  1/36
      EN1= -1.
      EN2=  1.
   ELSE IF (N.EQ.14) THEN
      IGROUP= 1
      C= -.444444444444      ! = -8/18
      EN1= -1.
      EN2=  .5
   ELSE IF (N.EQ.15) THEN
      IGROUP= 3
      C=  .6666666666667      ! =  2/3
      EN1= -1.
```

230

```
          EN2=  0.
ELSE IF (N.EQ.16) THEN
     IGROUP= 1
     C= -.444444444444      ! = -8/18
     EN1= -1.
     EN2= -.5
ELSE IF (N.EQ.17) THEN
     IGROUP= 1
     C= 7.111111111111      ! =  64/9
     EN1= -.5
     EN2= -.5
ELSE IF (N.EQ.18) THEN
     IGROUP= 2
     C=-10.666666666667      ! = -32/3
     EN1=  0.
     EN2= -.5
ELSE IF (N.EQ.19) THEN
     IGROUP= 1
     C= 7.111111111111      ! =  64/9
     EN1=  .5
     EN2= -.5
ELSE IF (N.EQ.20) THEN
     IGROUP= 3
     C=-10.666666666667      ! = -32/3
     EN1=  .5
     EN2=  0.
ELSE IF (N.EQ.21) THEN
     IGROUP= 1
     C= 7.111111111111      ! =  64/9
     EN1=  .5
     EN2=  .5
ELSE IF (N.EQ.22) THEN
     IGROUP= 2
     C=-10.666666666667      ! = -32/3
     EN1=  0.
     EN2=  .5
ELSE IF (N.EQ.23) THEN
     IGROUP= 1
     C= 7.111111111111      ! =  64/9
     EN1= -.5
     EN2=  .5
ELSE IF (N.EQ.24) THEN
     IGROUP= 3
     C=-10.666666666667      ! = -32/3
     EN1= -.5
     EN2=  0.
ELSE IF (N.EQ.25) THEN
     IGROUP= 4
     C= 16.
     EN1=  0.
     EN2=  0.
END IF

  IF (IGROUP.EQ.1) THEN
     X= C*(4.*EN1**2*ET1**4+ 4.*EN1**3*ET1**3- ET1**2- EN1*ET1)*
 >         (4.*EN2**2*ET2**4+ 4.*EN2**3*ET2**3- ET2**2- EN2*ET2)
   ELSE IF (IGROUP.EQ.2) THEN
     X= C*(ET1**4- 1.25*ET1**2+ .25)*
```

```
    >           (4.*EN2**2*ET2**4+ 4.*EN2**3*ET2**3- ET2**2- EN2*ET2)
      ELSE IF (IGROUP.EQ.3) THEN
        X= C*(4.*EN1**2*ET1**4+ 4.*EN1**3*ET1**3- ET1**2- EN1*ET1)*
    >           (ET2**4- 1.25*ET2**2+ .25)
      ELSE IF (IGROUP.EQ.4) THEN
        X= C*(ET1**4- 1.25*ET1**2+ .25)*(ET2**4- 1.25*ET2**2+ .25)
      END IF
      PHI4= X
      RETURN
      END
```

T.   Subroutine GAUSS

```
      SUBROUTINE GAUSS(N)

C*********************************************************************
C Solves for X matrix, where A*X=B, by applying Gauss-Jordan elimination
C*********************************************************************

      PARAMETER (MXE= 25,        MYE= 20) ! # elements in x & y dirs
      PARAMETER (MTC1= (1+MXE)*(1+MYE))   ! # corner nodes
      PARAMETER (MTC2= 2*(1+MYE)+1+MXE+2) ! extra nodes for mixed press
      PARAMETER (MTC= MTC1+MTC2)          ! total # press nodes

      COMMON/MXTRAN/
     >AMAT(MTC,MTC),BVEC(MTC),XVEC(MTC)

C Forward Elimination

      NM1= N-1
      DO 1 I= 1,NM1
        IP1= I+1
        DO 1 J= IP1,N
          C= AMAT(J,I)/AMAT(I,I)
          DO 2 K= IP1,N
            AMAT(J,K)= AMAT(J,K)- C*AMAT(I,K)
    2     CONTINUE
          BVEC(J)= BVEC(J)- C*BVEC(I)
    1 CONTINUE

C Back elimination. Reduces A to the identity matrix,
C while B becomes the solution vector.

      DO 10 I= 1,N
        NMI= N-I+1
        BVEC(NMI)= BVEC(NMI)/AMAT(NMI,NMI)
        XVEC(NMI)= BVEC(NMI)
        IF (NMI.EQ.1) GOTO 20
        NM1= NMI-1
        DO 10 J= 1,NM1
          NMJ= NMI-J
          BVEC(NMJ)= BVEC(NMJ)-AMAT(NMJ,NMI)*BVEC(NMI)
   10 CONTINUE
   20 CONTINUE
C
      RETURN
```

```
                END



    U.    Subroutine MENISCUS

             SUBROUTINE MENISCUS (CANG,BO,CA,MA,PR,BI,RS,PREF,RASPECT,PVAP,
             >          YREF,CR,FRHO,VREC,NINT,NPRINT,ISYMM,ICURVE,ITYPE,
             >          IPRESS)

C****************************************************************************
C Solves for both sides of meniscus between two parallel plates
C****************************************************************************

             IMPLICIT DOUBLE PRECISION (A-H)
             IMPLICIT DOUBLE PRECISION (O-Z)

             PARAMETER (MXE= 25,        MYE= 20) ! # elements in x & y direcs
             PARAMETER (MXN= 1+2*MXE)            ! # nodes in x-direction

             COMMON/MENVAR/
           * A(3,MXE),B(3,MXE),C(3,MXE),AP(2),AQ(2)

             COMMON/SURFACE/
           *XP(MXN),YP(MXN),YSURF0(MXN),PD(MXN),PB(MXN),TEMP(MXN),
           *SURFNN(MXN,2),SURFNT(MXN,2)

             COMMON/PLOT/
           *IMODE,IOUT8,IOUT9

             DOUBLE PRECISION MA

             DATA TOLL,TOLA/.0001,.00001/ ! Length and angle conv tolerances
             DATA MAXSIT/100000/          ! Maximum number of s iterations
             DATA MAXNEWT/300/            ! Maximum number of N-R iterations
             DATA MAXEND/300/             ! Maximum number of endpoint its
             DATA MAXANG/300/
             DATA IMETHOD/1/                 ! Sol method.=0 secant/=1 biscection
             DATA E/2.718281828/

             ANGOB= 1.5707963-CANG/57.2957795
             DELS= 1./DBLE(REAL(NINT))

C Surface scalar interpolation coefficients. 2nd order for temperature.
C 1st order for pressure when IPRESS=0 and 2nd order in corners when
C IPRESS=1 (or P=Ax2+Bx+C)

             NE= 0
             DO 10 N= 3,NPRINT,2
             NE= NE+1
             N1= N
             N2= N-1
             N3= N-2
             X1= XP(N1)
             X2= XP(N2)
             X3= XP(N3)
             DO 10 K= 1,2
               IF (K.EQ.1) THEN
```

```
              VAR1= PD(N1)
              VAR2= PD(N2)
              VAR3= PD(N3)
              IF (NE.EQ.1.OR.NE.EQ.(NPRINT-1)/2.AND.IPRESS.EQ.1) THEN
                A(K,NE)=((VAR1-VAR2)/(X1-X2)-(VAR2-VAR3)/(X2-X3))/(X1-X3)
                B(K,NE)=(VAR1-VAR2)/(X1-X2)-A(K,NE)*(X1+X2)
                C(K,NE)=VAR1-A(K,NE)*X1**2-B(K,NE)*X1
              ELSE
                A(K,NE)=0.
                B(K,NE)=(VAR1-VAR3)/(X1-X3)
                C(K,NE)=VAR1-B(K,NE)*X1
              END IF
            ELSE IF (K.EQ.2) THEN
              VAR1= TEMP(N1)
              VAR2= TEMP(N2)
              VAR3= TEMP(N3)
              A(K,NE)= ((VAR1-VAR2)/(X1-X2)-(VAR2-VAR3)/(X2-X3))/(X1-X3)
              B(K,NE)= (VAR1-VAR2)/(X1-X2)-A(K,NE)*(X1+X2)
              C(K,NE)= VAR1-A(K,NE)*X1**2-B(K,NE)*X1
            END IF
       10 CONTINUE

C Check if exponential temperature approximation necessary for end
C surface elements

          DO 20 N= 1,2
          AP(N)= 0.
          IF (N.EQ.1) THEN
            X1= XP(3)
            X2= XP(2)
            X3= XP(1)
            V1= TEMP(3)
            V2= TEMP(2)
            V3= TEMP(1)
            NE= 1
            NE0= 2
          ELSE
            X1= XP(NPRINT-2)
            X2= XP(NPRINT-1)
            X3= XP(NPRINT)
            V1= TEMP(NPRINT-2)
            V2= TEMP(NPRINT-1)
            V3= TEMP(NPRINT)
            NE= (NPRINT-1)/2
            NE0= NE-1
          END IF
          DEL= X3-X1
          DX= DEL/100.
          FMIN= A(2,NE)*X1**2+B(2,NE)*X1+C(2,NE)
          FMID= A(2,NE)*X2**2+B(2,NE)*X2+C(2,NE)
          FMAX= A(2,NE)*X3**2+B(2,NE)*X3+C(2,NE)

          XT= X1
          DO 21 K= 2,100
            XT= XT+DX
            F= A(2,NE)*XT**2+B(2,NE)*XT+C(2,NE)
            IF ((XT-X1)/DEL.LE.0.5) THEN
              IF (ITYPE.EQ.0) THEN
```

```fortran
            IF (F.LT.FMIN.OR.F.GT.FMID) GOTO 22
          ELSE
            IF (F.GT.FMIN.OR.F.LT.FMID) GOTO 22
          END IF
        ELSE
          IF (ITYPE.EQ.0) THEN
            IF (F.LT.FMID.OR.F.GT.FMAX) GOTO 22
          ELSE
            IF (F.GT.FMID.OR.F.LT.FMAX) GOTO 22
          END IF
        END IF
21    CONTINUE
      GOTO 20
22    CONTINUE

      AP(N)= .5
      IF (N.EQ.1) THEN
        XA= 1.-X1
        XB= 1.-X2
        XC= 1.-X3
      ELSE
        XA= X1
        XB= X2
        XC= X3
      END IF
      IF (ITYPE.EQ.1) THEN
        V1= 1.-V1
        V2= 1.-V2
        V3= 1.-V3
        AP(N)= 1.
      END IF
      V0MAX= V1
      V0MIN= V0MAX
      DELV= .001
      DO 23 K= 1,100000
        V0MIN= V0MIN-DELV
        FX= FUNC2(V0MIN,XA,XB,XC,V1,V2,V3)
        IF (FX.GT.0.) GOTO 24
23    CONTINUE
      STOP
24    CONTINUE
      V0= V0MIN
      DO 25 K= 1,10000
        V00= V0
        V0= (V0MIN+V0MAX)/2.
        FX= FUNC2(V0,XA,XB,XC,V1,V2,V3)
        IF (FX.LE.0.) THEN
          V0MAX= V0
        ELSE
          V0MIN= V0
        END IF
        IF (ABS((V0-V00)/V00).LE..00001) GOTO 26
25    CONTINUE
      STOP
26    CONTINUE

      X0= XC*DLOG(V2-V0)-XB*DLOG(V3-V0)
      X0= X0/(DLOG(V2-V0)-DLOG(V3-V0))
```

```
        COEFF= DLOG(V3-V0)/(XC-X0)
        C(2,NE)= V0
        IF (N.EQ.1) THEN
          A(2,NE)= -1.*COEFF
          B(2,NE)= -1.*(X0-1.)
        ELSE
          A(2,NE)= COEFF
          B(2,NE)= X0
        END IF

  20 CONTINUE

C Check if exponential press fit necessary for end surface elements

        IF (IPRESS.EQ.1.AND.ICURVE.NE.0) THEN
        DO 30 N= 1,2
        AQ(N)= 0.
        IF (N.EQ.1) THEN
          X1= XP(3)
          X2= XP(2)
          X3= XP(1)
          V1= PD(3)
          V2= PD(2)
          V3= PD(1)
          NE= 1
          NE0= 2
        ELSE
          X1= XP(NPRINT-2)
          X2= XP(NPRINT-1)
          X3= XP(NPRINT)
          V1= PD(NPRINT-2)
          V2= PD(NPRINT-1)
          V3= PD(NPRINT)
          NE= (NPRINT-1)/2
          NE0= NE-1
        END IF
        DEL= X3-X1
        DX= DEL/100.
        FMIN= A(1,NE)*X1**2+B(1,NE)*X1+C(1,NE)
        FMID= A(1,NE)*X2**2+B(1,NE)*X2+C(1,NE)
        FMAX= A(1,NE)*X3**2+B(1,NE)*X3+C(1,NE)
          IF (FMAX.LT.FMID.AND.FMID.GE.FMIN) GOTO 30
          IF (FMAX.GT.FMID.AND.FMID.LE.FMIN) GOTO 30

        XT= X1
        DO 31 K= 2,100
          XT= XT+DX
          F= A(1,NE)*XT**2+B(1,NE)*XT+C(1,NE)
          IF ((XT-X1)/DEL.LE.0.5) THEN
        ·    IF (ITYPE.EQ.1) THEN
              IF (F.LT.FMIN) GOTO 32 ! Cond
            ELSE
              IF (F.GT.FMIN) GOTO 32 ! Evap
            END IF
          END IF
  31    CONTINUE
        GOTO 30
  32    CONTINUE
```

236

```fortran
      AQ(N)= .5
      IF (N.EQ.1) THEN
        XA= 1.-X1
        XB= 1.-X2
        XC= 1.-X3
      ELSE
        XA= X1
        XB= X2
        XC= X3
      END IF
      IF (ITYPE.EQ.0) THEN
        V1= -1.*V1
        V2= -1.*V2
        V3= -1.*V3
        AQ(N)= 1.
      END IF
      V0MAX= V1
      V0MIN= V0MAX
      DELV= .001
      DO 33 K= 1,10000000
        V0MIN= V0MIN-DELV
        FX= FUNC2(V0MIN,XA,XB,XC,V1,V2,V3)
        IF (FX.GT.0.) GOTO 34
33    CONTINUE
      STOP
34    CONTINUE
      V0= V0MIN
      DO 35 K= 1,10000000
        V00= V0
        V0= (V0MIN+V0MAX)/2.
        FX= FUNC2(V0,XA,XB,XC,V1,V2,V3)
        IF (FX.LE.0.) THEN
          V0MAX= V0
        ELSE
          V0MIN= V0
        END IF
        IF (ABS((V0-V00)/V00).LE..0000001) GOTO 36
35    CONTINUE
      STOP
36    CONTINUE

      X0= XC*DLOG(V2-V0)-XB*DLOG(V3-V0)
      X0= X0/(DLOG(V2-V0)-DLOG(V3-V0))
      COEFF= DLOG(V3-V0)/(XC-X0)
      C(1,NE)= V0
      IF (N.EQ.1) THEN
        A(1,NE)= -1.*COEFF
        B(1,NE)= -1.*(X0-1.)
      ELSE
        A(1,NE)= COEFF
        B(1,NE)= X0
      END IF

30 CONTINUE
      END IF

C Max curvature point variance iteration loop
```

```
         XM= 0.5
         XP((NPRINT+1)/2)= .5
         XLR= 0.
         XOBJ= .5
         ANGLIM= 0.

         DO 210 NPASS= 1,20

C Max and min curvature calculations

         IF (ITYPE.EQ.0) THEN
           FTEMP= FUNC(XM,2,NPRINT)
         ELSE IF (ITYPE.EQ.1) THEN
           FTEMP= FUNC(XM,2,NPRINT)-1.
         END IF

         IF (ICURVE.EQ.0) THEN           ! Bo only
           TMIN= 0.
           TMAX= 2.*DCOS(CANG/57.2957795)
         ELSE IF (ICURVE.EQ.1) THEN      ! Bo and Pd
           TMIN= CA*FUNC(XM,1,NPRINT)
           TMAX= 2.*DCOS(CANG/57.2957795)+TMIN
         ELSE IF (ICURVE.EQ.2) THEN      ! Bo and Vr
           TMIN= -1.*VREC*FTEMP**2
           TMAX= 2.*DCOS(CANG/57.2957795)+TMIN
         ELSE IF (ICURVE.EQ.3) THEN      ! Bo and Cr
           TMIN= 0.
           TMAX= 2.*DCOS(CANG/57.2957795)*
     >          (1.-CR*FUNC(XM,2,NPRINT))+TMIN
         ELSE IF (ICURVE.EQ.4) THEN      ! Bo, Pd and Vr
           TMIN= CA*FUNC(XM,1,NPRINT)
     >          -VREC*FTEMP**2
           TMAX= 2.*DCOS(CANG/57.2957795)+TMIN
         ELSE IF (ICURVE.EQ.5) THEN      ! Bo, Pd and Cr
           TMIN= CA*FUNC(XM,1,NPRINT)
           TMAX= 2.*DCOS(CANG/57.2957795)*
     >          (1.-CR*FUNC(XM,2,NPRINT))+TMIN
         ELSE IF (ICURVE.EQ.6) THEN      ! Bo, Vr and Cr
           TMIN= -1.*VREC*FTEMP**2
           TMAX= 2.*DCOS(CANG/57.2957795)*
     >          (1.-CR*FUNC(XM,2,NPRINT))+TMIN
         ELSE IF (ICURVE.EQ.7) THEN      ! Bo, Pd, Vr and Cr
           TMIN= CA*FUNC(XM,1,NPRINT)
     >          -VREC*FTEMP**2
           TMAX= 2.*DCOS(CANG/57.2957795)*
     >          (1.-CR*FUNC(XM,2,NPRINT))+TMIN
         END IF

         T0= TMAX
         T1= TMIN
         IF (NPASS.GE.2) THEN
           IF (ICLOSE.EQ.1) THEN
             IF (NPASS.EQ.2) ANGLIM= -1.5707963
             TMAX= (1.+.1*REAL(NPASS-1))*TMAX
             T0= TMAX
           ELSE
             IF (NPASS.EQ.2) ANGLIM= 1.5707963
```

```fortran
              FAC= 1.+.1*REAL(NPASS-2)
              TMIN= (1.+.1*REAL(NPASS-1))*(1.+1.2*FAC)*TMIN-1.2*FAC*TMAX
              T1= TMIN
            END IF
          END IF


          XA= XM
          XB= 0.


        IF (NPASS.EQ.1.OR.(NPASS.GE.2.AND.ICLOSE.EQ.1)) THEN
        CALL CURVE
      >   (XA,XB,ANGOB,ANGWMAX,DELS,XLL,XPL,TMAX,BO,CA,MA,PR,BI,RS,
      >    PVAP,CR,FRHO,VREC,NPRINT,ICURVE,ITYPE)

        IF (IMODE.EQ.1) WRITE(6,1000)NPASS,XLL,T,TMAX,TMAX,ANGWMAX
      >*57.2957795,(1.5707963-ANGWMAX)*57.2957795
        IF (IOUT8.GE.2) WRITE(8,1000)NPASS,XTOT,T,TMAX,TMAX,ANGWMAX
      >*57.2957795,(1.5707963-ANGWMAX)*57.2957795
        END IF


        IF (NPASS.EQ.1.OR.(NPASS.GE.2.AND.ICLOSE.EQ.0)) THEN
        CALL CURVE
      >   (XA,XB,ANGOB,ANGWMIN,DELS,XLL,XPL,TMIN,BO,CA,MA,PR,BI,RS,
      >    PVAP,CR,FRHO,VREC,NPRINT,ICURVE,ITYPE)

        IF (IMODE.EQ.1) WRITE(6,1000)NPASS,XLL,T,TMIN,TMIN,ANGWMIN
      >*57.2957795,(1.5707963-ANGWMIN)*57.2957795
        IF (IOUT8.GE.2) WRITE(8,1000)NPASS,XTOT,T,TMIN,TMIN,ANGWMIN
      >*57.2957795,(1.5707963-ANGWMIN)*57.2957795
        END IF


        IF (ANGOB.GT.ANGWMAX) THEN
          ICLOSE= 1   ! Middle bulge down
        ELSE IF (ANGOB.LT.ANGWMIN) THEN
          ICLOSE= 0   ! Middle bulge up
        ELSE
          GOTO 205
        END IF

  210 CONTINUE
      STOP
  205 CONTINUE

      NITTER= 0
      DO 200 J= 1,25

        T= (T0+T1)/2.
      ANGW01= ANGW

C Left hand side of meniscus

      XA= XM
      XB= 0.
      CALL CURVE
      >   (XA,XB,ANGOB,ANGW,DELS,XLL,XPL,T,BO,CA,MA,PR,BI,RS,PVAP,CR,
      >    FRHO,VREC,NPRINT,ICURVE,ITYPE)

      XTOT= XLR+XLL
```

```fortran
C Check length

      IF (IMODE.EQ.1) WRITE(6,1000)J,XTOT,T,TMIN,TMAX,ANGW
     >  *57.2957795,(1.5707963-ANGW)*57.2957795
      IF (IOUT8.GE.2) WRITE(8,1000)J,XTOT,T,TMIN,TMAX,ANGW
     >  *57.2957795,(1.5707963-ANGW)*57.2957795

        IF (NPASS.GE.2) THEN
          IF (ICLOSE.EQ.1) THEN
            IF (ANGW.LT.1.5707963.AND.ANGW.GT.ANGLIM) ANGLIM= ANGW
          ELSE IF (ICLOSE.EQ.0) THEN
            IF (ANGW.GT.-1.5707963.AND.ANGW.LT.ANGLIM) ANGLIM= ANGW
          END IF
        END IF

C Curvature correction

      IF (DABS(ANGW-ANGOB).LE.TOLL) GOTO 300
      IF (ANGW.LT.ANGOB.OR.ANGW.LE.-1.5707963) THEN
        T1= T
      ELSE IF (ANGW.GT.ANGOB.OR.ANGW.GE.1.5707963) THEN
        T0= T
      END IF
      IF (NITTER.GE.8) GOTO 201

  200 CONTINUE
  201 CONTINUE

      IF (ICLOSE.EQ.1) THEN
      IF (IMODE.EQ.1) WRITE(6,2000)ANGLIM*57.2957795
      IF (IOUT8.GE.2) WRITE(8,2000)ANGLIM*57.2957795
      ELSE
      IF (IMODE.EQ.1) WRITE(6,2001)ANGLIM*57.2957795
      IF (IOUT8.GE.2) WRITE(8,2001)ANGLIM*57.2957795
      END IF

  300 CONTINUE

      Y0= RASPECT-YP(1)

      DX= XP(2)-XP(1)
      DO 510 K= 1,NPRINT
        YP(K)= YP(K)+Y0
      N= K
      IF (DABS(XP(K)-.5).LE..1*DX) GOTO 511
  510 CONTINUE
  511 CONTINUE
      DO 520 KK= N+1,NPRINT
        YP(KK)= YP(NPRINT+1-KK)
        PD(KK)= PD(NPRINT+1-KK)
        PB(KK)= PB(NPRINT+1-KK)
        TEMP(KK)= TEMP(NPRINT+1-KK)
        SURFNN(KK,1)= -1.*SURFNN(NPRINT+1-KK,1)
        SURFNN(KK,2)= SURFNN(NPRINT+1-KK,2)
        SURFNT(KK,1)= SURFNT(NPRINT+1-KK,1)
        SURFNT(KK,2)= -1.*SURFNT(NPRINT+1-KK,2)
  520 CONTINUE
```

240

```
         YREF= YP(1)

1000 FORMAT(/,' CURV ITERATION # (J)= ',13X,I3,/,
     *           ' Calced Length=    ',F16.10,/,
     *           ' T=                ',F16.10,/,
     *           ' Tmin=             ',F16.10,/,
     *           ' Tmax=             ',F16.10,/,
     *           ' Contour Angle=    ',F16.10,/,
     *           ' Contact Angle=    ',F16.10)
2000 FORMAT(' Meniscus curvature failed to maximize to desired',/,
     >           ' contact angle. Maximum angle reached= ',E12.6)
2001 FORMAT(' Meniscus curvature failed to minimize to desired',/,
     >           ' contact angle. Minimum angle reached= ',E12.6)

     RETURN
     END
```

V.   Subroutine CURVE

```
     SUBROUTINE CURVE
     >  (XA,XB,ANGOB,ANGW,DELS,XLENGTH,XREAL,T,BO,CA,MA,PR,BI,RS,
     >  PVAP,CR,FRHO,VREC,NPRINT,ICURVE,ITYPE)

     IMPLICIT DOUBLE PRECISION (A-H)
     IMPLICIT DOUBLE PRECISION (O-Z)

     PARAMETER (MXE= 25,        MYE= 20) ! # elements in x & y direcs
     PARAMETER (MXN= 1+2*MXE)             ! # nodes in x-direction

     COMMON/MENVAR/
     * A(3,MXE),B(3,MXE),C(3,MXE),AP(2),AQ(2)

     COMMON/SURFACE/
     *XP(MXN),YP(MXN),YSURF0(MXN),PD(MXN),PB(MXN),TEMP(MXN),
     *SURFNN(MXN,2),SURFNT(MXN,2)

     COMMON/PLOT/
     *IMODE,IOUT8,IOUT9

     DOUBLE PRECISION MA

     DATA MAXS,MAXA,MAXL,IERROR/4000000,300,300,0/
     DATA TOLL,TOLA/.00001,.0000001/

     IF (XA.LT.XB) THEN
       FAC= 1.
       IFAC= 1
       XOBJ= 1.
     ELSE
       FAC= -1.
       IFAC= -1
       XOBJ= 0.
     END IF

     DO 10 N= 1,NPRINT
       IF (XP(N).EQ.XA) THEN
```

```
        YP(N)= 0.
        SURFNN(N,1)= 0.
        SURFNN(N,2)= 1.
        SURFNT(N,1)= 1.
        SURFNT(N,2)= 0.
        IF (FAC.GT.0.) THEN
          NFLAG= N+1
        ELSE
          NFLAG= N-1
        END IF
        GOTO 11
      ELSE IF (XP(N).GT.XA) THEN
        IF (FAC.GT.0.) THEN
          NFLAG= N
        ELSE
          NFLAG= N-1
        END IF
        GOTO 11
      END IF
10 CONTINUE
11 CONTINUE
   NMID= N

   S1= 0.
   S0= 0.
   SM= 0.
   SM1= 0.
   X1= 0.
   XREAL= XA
   Y1= 0.
   ALPH1= 0.                          ! Angle at i-1
   ALPH0= 0.                          ! Angle at i-2
   ALPHM= 0.                          ! Angle at i-3
   ALPNHM1= 0.                        ! Angle at i-4

   DO 100 N= 2,MAXS
     S2= S1+DELS
   ALPH2= ALPH1

   XREAL1= XREAL
   XREAL0= XREAL1

   DO 110 L= 1,MAXL

   IF (ITYPE.EQ.0) THEN
     FTEMP= FUNC(XREAL1,2,NPRINT)
   ELSE IF (ITYPE.EQ.1) THEN
     FTEMP= FUNC(XREAL1,2,NPRINT)-1.
   END IF

     IF (ICURVE.EQ.0) THEN           ! Bo only
   CON1= T+BO*Y1
   CON2= 1.
   ELSE IF (ICURVE.EQ.1) THEN     ! Bo and Pd
   CON1=T+BO*Y1-CA*FUNC(XREAL1,1,NPRINT)
   CON2=1
   ELSE IF (ICURVE.EQ.2) THEN     ! Bo and Vr
   CON1= T+BO*Y1+VREC*FTEMP**2
```

```
      CON2= 1.
     ELSE IF (ICURVE.EQ.3) THEN      ! Bo and Cr
      CON1= T+BO*Y1
      CON2= 1.-CR*FUNC(XREAL1,2,NPRINT)
     ELSE IF (ICURVE.EQ.4) THEN      ! Bo, Pd and Vr
      CON1= T+BO*Y1-CA*FUNC(XREAL1,1,NPRINT)
    >          +VREC*FTEMP**2
      CON2= 1.
     ELSE IF (ICURVE.EQ.5) THEN      ! Bo, Pd and Cr
      CON1=T+BO*Y1-CA*FUNC(XREAL1,1,NPRINT)
      CON2= 1.-CR*FUNC(XREAL1,2,NPRINT)
     ELSE IF (ICURVE.EQ.6) THEN      ! Bo, Vr and Cr
      CON1= T+BO*Y1+VREC*FTEMP**2
      CON2= 1.-CR*FUNC(XREAL1,2,NPRINT)
     ELSE IF (ICURVE.EQ.7) THEN      ! Bo, Pd, Vr and Cr
      CON1= T+BO*Y1-CA*FUNC(XREAL1,1,NPRINT)
    >          +VREC*FTEMP**2
      CON2= 1.-CR*FUNC(XREAL1,2,NPRINT)
     END IF

         IF (N.EQ.2.AND.L.EQ.1) THEN
           RC1= CON1/CON2
           IF (XP(NMID).EQ.XA) PB(NMID)= .5/DSIN(ANGOB)*
    >        ((1.-CR*FUNC(XP(NMID),2,NPRINT))*RC1+VREC*FTEMP**2)
         END IF
       ANGNEW= 0.
       DO 120 M= 1,MAXA
         IF (N.EQ.2) THEN
           ANGNEW= ALPH1+(CON1*DELS+BO*(DSIN(ALPH1)+DSIN(ALPH2))
    >              /2.*DELS**2)/CON2
         ELSE
           ANG= (3.*ALPH2+6.*ALPH1-ALPH0)/8.
           ANGNEW= (4.*ALPH1-ALPH0)/3.+
    >              (2.*CON1*DELS+BO*DELS**2/3.*
    >              (DSIN(ALPH1)+4.*DSIN(ANG)+DSIN(ALPH2)))/
    >              3./CON2
         END IF
         IF (DABS(ANGNEW-ALPH2).LE.TOLA) GOTO 130
         ALPH2= ANGNEW
 120     CONTINUE

       IERROR= 1
       IF (IOUT8.NE.0) WRITE(8,1000)
       IF (IMODE.EQ.1) WRITE(6,1000)
 1000    FORMAT(' ANGLE ITERATION DID NOT CONVERGE')
       IF (IERROR.EQ.1) GOTO 9999

 130     CONTINUE
       IF (ANGNEW.LE..001.AND.N.EQ.2) THEN
           ANGNEW= (ALPH1*(CON2+BO/2.*DELS**2)+CON1*DELS)/
    >              (CON2-BO/2.*DELS**2)
         ELSE IF (ANGNEW.LE..001.AND.N.NE.2) THEN
         ANGNEW= 4.*ALPH1-ALPH0+BO*DELS**2/6./CON2*
    >              (8.*ALPH1-ALPH0)+2.*CON1*DELS/CON2
         ANGNEW= ANGNEW/(3.-5.*BO*DELS**2/6./CON2)
       END IF
       ALPH2= ANGNEW
```

```fortran
      IF (N.EQ.2) THEN
        XNEW= X1+(DCOS(ALPH1)+DCOS(ALPH2))/2.*DELS
        YNEW= Y1+(DSIN(ALPH1)+DSIN(ALPH2))/2.*DELS
      ELSE
        ANG= (3.*ALPH2+6.*ALPH1-ALPH0)/8.
        XNEW= X1+(DCOS(ALPH1)+DCOS(ALPH2)+ 4.*DCOS(ANG))*DELS/6.
        YNEW= Y1+(DSIN(ALPH1)+DSIN(ALPH2)+ 4.*DSIN(ANG))*DELS/6.
      END IF
      XREAL1= XREAL+FAC*(XNEW-X1)
      IF (DABS(XREAL1-XREAL0).LE.TOLL) GOTO 140
      XREAL0= XREAL1
  110 CONTINUE

      IERROR= 1
      IF (IOUT8.NE.0) WRITE(8,1001)
      IF (IMODE.EQ.1) WRITE(6,1001)
 1001 FORMAT(' LENGTH ITERATION DID NOT CONVERGE')
      IF (IERROR.EQ.1) GOTO 9999
C
  140 CONTINUE

      IF (N.EQ.2) THEN
        RC2= (ALPH2-ALPH1)/DELS
      ELSE
        RC2= (3.*ALPH2-4.*ALPH1+ALPH0)/2./DELS
      END IF
      X2= XNEW
      Y2= YNEW
      XREAL1= XREAL+FAC*(XNEW-X1)
      IF (XREAL1.LT.0..OR.XREAL1.GT.1.) THEN

      FPRESS= FUNC(XOBJ,1,NPRINT)
      IF (ITYPE.EQ.0) THEN
        FTEMP= FUNC(XOBJ,2,NPRINT)
      ELSE IF (ITYPE.EQ.1) THEN
        FTEMP= FUNC(XOBJ,2,NPRINT)-1.
      END IF

      IF (ICURVE.EQ.0) THEN           ! Bo only
        CON1= T+BO*Y1
        CON2= 1.
      ELSE IF (ICURVE.EQ.1) THEN      ! Bo and Pd
        CON1=T+BO*Y1-CA*FPRESS
        CON2=1
      ELSE IF (ICURVE.EQ.2) THEN      ! Bo and Vr
        CON1= T+BO*Y1+VREC*FTEMP**2
        CON2= 1.
      ELSE IF (ICURVE.EQ.3) THEN      ! Bo and Cr
        CON1= T+BO*Y1
        CON2= 1.-CR*FUNC(XOBJ,2,NPRINT)
      ELSE IF (ICURVE.EQ.4) THEN      ! Bo, Pd and Vr
        CON1= T+BO*Y1-CA*FPRESS+VREC*FTEMP**2
        CON2= 1.
      ELSE IF (ICURVE.EQ.5) THEN      ! Bo, Pd and Cr
        CON1=T+BO*Y1-CA*FPRESS
        CON2= 1.-CR*FUNC(XOBJ,2,NPRINT)
      ELSE IF (ICURVE.EQ.6) THEN      ! Bo, Vr and Cr
        CON1= T+BO*Y1+VREC*FTEMP**2
```

```fortran
              CON2= 1.-CR*FUNC(XOBJ,2,NPRINT)
          ELSE IF (ICURVE.EQ.7) THEN      ! Bo, Pd, Vr and Cr
              CON1= T+BO*Y1-CA*FPRESS+VREC*FTEMP**2
              CON2= 1.-CR*FUNC(XOBJ,2,NPRINT)
          END IF

          DSMIN= 0.
          DSMAX= DELS
          DS2= DELS
          DO 400 L= 1,MAXL
            DS1= .5*(DSMIN+DSMAX)
            DO 401 M= 1,MAXA
              ANGNEW= ALPH2
              ANG= (2.*DS2+DS1)/DS2/4.*ALPH1+
     >              (2.*DS2+DS1)/(DS1+DS2)/4.*ANGNEW-
     >              DS1/DS2*DS1/(DS1+DS2)/4.*ALPH0
              ALPH2= DS1/CON2*(DS1+DS2)/(2.*DS1+DS2)*
     >          (CON1+BO*DS1/6.*(DSIN(ALPH2)+4.*DSIN(ANG)+DSIN(ALPH1)))+
     >          (DS1+DS2)/DS2*(DS1+DS2)/(2.*DS1+DS2)*ALPH1-
     >          DS1/DS2*DS1/(2.*DS1+DS2)*ALPH0
              IF (DABS(ANGNEW-ALPH2).LE.TOLA) GOTO 402
401         CONTINUE
402         CONTINUE
            ANG= (2.*DS2+DS1)/DS2/4.*ALPH1+
     >            (2.*DS2+DS1)/(DS1+DS2)/4.*ALPH2-
     >            DS1/DS2*DS1/(DS1+DS2)/4.*ALPH0
            XNEW= X1+(DCOS(ALPH1)+DCOS(ALPH2)+ 4.*DCOS(ANG))*DS1/6.
            YNEW= Y1+(DSIN(ALPH1)+DSIN(ALPH2)+ 4.*DSIN(ANG))*DS1/6.
            XREAL1= XREAL+FAC*(XNEW-X1)
            IF (DABS(XREAL1-XOBJ).LE.TOLL/1000.) GOTO 403
            IF ((XREAL1.LT.XOBJ.AND.FAC.LT.0.).OR.
     >          (XREAL1.GT.XOBJ.AND.FAC.GT.0.)) THEN
              DSMAX= DS1
            ELSE
              DSMIN= DS1
            END IF
400       CONTINUE
403       CONTINUE
          X2= XNEW
          Y2= YNEW
          ANGW= ALPH2
          ALPHA= ALPH2
          GOTO 300
        END IF

        IF (DABS(ALPH2).GT.1.5707963) THEN
          ANGW= ALPH2
          GOTO 300
        ELSE
        X2= XNEW
        Y2= YNEW
        XREAL= XREAL+FAC*(X2-X1)
        NOLD= NFLAG
        IF (XREAL.EQ.XP(NFLAG)) THEN
          YP(NFLAG)= YNEW
          ALPHA= ALPH2
            RCURV= RC2
          IF (FAC.GT.0.) NFLAG= NFLAG+1
```

```
                IF (FAC.LT.0.) NFLAG= NFLAG-1
              ELSE IF (XREAL.GT.XP(NFLAG).AND.FAC.GT.0.) THEN
                DX= X2-X1-(XREAL-XP(NFLAG))
                YP(NFLAG)= (YNEW-Y1)/(XNEW-X1)*DX+Y1
                ALPHA= (ALPH2-ALPH1)/(XNEW-X1)*DX+ALPH1
                  RCURV= (RC2-RC1)/(XNEW-X1)*DX+RC1
                NFLAG= NFLAG+1
                ELSE IF (XREAL.LT.XP(NFLAG).AND.FAC.LT.0.) THEN
                DX= X2-X1-(XP(NFLAG)-XREAL)
                YP(NFLAG)= (YNEW-Y1)/(XNEW-X1)*DX+Y1
                ALPHA= (ALPH2-ALPH1)/(XNEW-X1)*DX+ALPH1
                  RCURV= (RC2-RC1)/(XNEW-X1)*DX+RC1
                NFLAG= NFLAG-1
                END IF
                PB(NOLD)= .5/DSIN(ANGOB)*
     >          ((1.-CR*FUNC(XREAL,2,NPRINT))*RCURV+VREC*FTEMP**2)
                SURFNT(NOLD,1)= DCOS(FAC*ALPHA)
              SURFNT(NOLD,2)= DSIN(FAC*ALPHA)
              SURFNN(NOLD,1)= -1.*SURFNT(NOLD,2)
              SURFNN(NOLD,2)= SURFNT(NOLD,1)

            END IF
C
          ALPHM1= ALPHM
          ALPHM= ALPH0
          ALPH0= ALPH1
          ALPH1= ALPH2
          SM1= SM
          SM= S0
          S0= S1
          S1= S2
          X1= X2
          Y1= Y2
            RC1= RC2

     100 CONTINUE

          IERROR= 1
          IF (IOUT8.NE.0) WRITE(8,1002)
          IF (IMODE.EQ.1) WRITE(6,1002)
    1002 FORMAT(' S INTEGRATION FAILED TO REACH ALPH=ANGOB')
          IF (IERROR.EQ.1) GOTO 9999

     300 CONTINUE
          X2= XNEW
          Y2= YNEW
          XREAL= XREAL+FAC*(X2-X1)

          PB(NFLAG)= .5/DSIN(ANGOB)*
     >       ((1.-CR*FUNC(XREAL,2,NPRINT))*RCURV+VREC*FTEMP**2)
          DX= X2-X1-FAC*(XREAL-XP(NFLAG))
          YP(NFLAG)= Y2
          SURFNT(NFLAG,1)= DCOS(FAC*ALPH2)
          SURFNT(NFLAG,2)= DSIN(FAC*ALPH2)
          SURFNN(NFLAG,1)= -1.*SURFNT(NFLAG,2)
          SURFNN(NFLAG,2)= SURFNT(NFLAG,1)

          XLENGTH= X2
```

246

```fortran
      IF (IMODE.EQ.1) WRITE(6,1003)XA,XB,DELS,XLENGTH,XREAL,T,BO,
     > NPRINT,NFLAG
      IF (IOUT8.GE.2) WRITE(8,1003)XA,XB,DELS,XLENGTH,XREAL,T,BO,
     > NPRINT,NFLAG
 1003 FORMAT(/,' XA=           ',F14.6,/,
     >          ' XB=           ',F14.6,/,
     >          ' DELS=         ',F14.6,/,
     >          ' XLENGTH=      ',F14.6,/,
     >          ' XREAL=        ',F14.6,/,
     >          ' T=            ',F14.6,/,
     >          ' BO=           ',F14.6,/,
     >          ' NPRINT=       ',I14,/,
     >          ' NFLAG=        ',I14)

 9999 CONTINUE
      IF (IERROR.EQ.1) STOP
C
      RETURN
      END
```

W.   Function FUNC

```fortran
      FUNCTION FUNC(X,I,NPRINT)

      IMPLICIT DOUBLE PRECISION (A-H)
      IMPLICIT DOUBLE PRECISION (O-Z)

      PARAMETER (MXE= 25,         MYE= 20) ! # elements in x & y direcs
      PARAMETER (MXN= 1+2*MXE)             ! # nodes in x-direction

      COMMON/MENVAR/
     * A(3,MXE),B(3,MXE),C(3,MXE),AP(2),AQ(2)

      COMMON/SURFACE/
     *XP(MXN),YP(MXN),YSURF0(MXN),PD(MXN),PB(MXN),TEMP(MXN),
     *SURFNN(MXN,2),SURFNT(MXN,2)

      DIMENSION ISOLVE1(2)

      INTEGER I,NPRINT,N,NEL,K,NE
      DATA E/2.718281828/

C ISOLVE1(N) = 0: FUNC set to 0 past walls
C            = 1: Linear decline from value at wall to 0 at DS
C            = 2: Constant at value at wall
C            = 3: Funcional extrapolation of variable value

      ISOLVE1(1)= 3          ! Pressure
      ISOLVE1(2)= 3          ! Temperature
      DS= .3

      IF (X.GE.XP(NPRINT-2)) THEN
      NEL= (NPRINT-1)/2
        IF (X.GT.XP(NPRINT)) THEN
          IF (ISOLVE1(I).EQ.3) THEN
```

```
          XT= X
        ELSE
          XT= XP(NPRINT)
        END IF
      ELSE
        XT= X
      END IF
  IF ((AP(2).GT..1.AND.I.EQ.2).OR.(AQ(2).GT..1.AND.I.EQ.1)) THEN
    ANS= E**(A(I,NEL)*(XT-B(I,NEL)))+C(I,NEL)
      IF (AP(2).GT..6.AND.I.EQ.2) ANS= 1.-ANS
      IF (AQ(2).GT..6.AND.I.EQ.1) ANS= -1.*ANS
  ELSE
    ANS= A(I,NEL)*XT**2+B(I,NEL)*XT+C(I,NEL)
  END IF
      IF (X.GT.XP(NPRINT)) THEN
        IF (ISOLVE1(I).EQ.0) THEN
          ANS= 0.
        ELSE IF (ISOLVE1(I).EQ.1) THEN
          IF (X.LE.XP(NPRINT)+DS) THEN
            ANS= ANS-ANS/DS*(X-XP(NPRINT))
          ELSE IF (X.GT.XP(NPRINT)+DS) THEN
            ANS= 0.
          END IF
        END IF
      END IF

ELSE IF (X.LE.XP(3)) THEN
NEL= 1
  IF (X.LT.XP(1)) THEN
      IF (ISOLVE1(I).EQ.3) THEN
        XT= X
      ELSE
        XT= XP(1)
      END IF
    ELSE
      XT= X
    END IF
  IF ((AP(1).GT..1.AND.I.EQ.2).OR.(AQ(1).GT..1.AND.I.EQ.1)) THEN
    ANS= E**(A(I,NEL)*(XT-B(I,NEL)))+C(I,NEL)
      IF (AP(1).GT..6.AND.I.EQ.2) ANS= 1.-ANS
      IF (AQ(1).GT..6.AND.I.EQ.1) ANS= -1.*ANS
  ELSE
    ANS= A(I,NEL)*XT**2+B(I,NEL)*XT+C(I,NEL)
  END IF
      IF (X.LT.XP(1)) THEN
        IF (ISOLVE1(I).EQ.0) THEN
          ANS= 0.
        ELSE IF (ISOLVE1(I).EQ.1) THEN
          IF (X.GE.XP(1)-DS) THEN
            ANS= ANS/DS*(X+DS-XP(1))
          ELSE IF (X.LT.XP(1)-DS) THEN
            ANS= 0.
          END IF
        END IF
      END IF

ELSE
NE= 1
```

```
      DO 1 K= 3,NPRINT-2,2
        NE= NE+1
        IF (X.GE.XP(K).AND.X.LE.XP(K+2)) THEN
          NEL= NE
          GOTO 2
        END IF
1     CONTINUE
2     CONTINUE
      ANS= A(I,NEL)*X**2+B(I,NEL)*X+C(I,NEL)
      END IF

      FUNC= ANS
      RETURN
      END
```

X.   Function FUNC2

```
      FUNCTION FUNC2(B,X1,X2,X3,V1,V2,V3)

      IMPLICIT DOUBLE PRECISION (A-H)
      IMPLICIT DOUBLE PRECISION (O-Z)

      F1= DLOG(V1-B)-DLOG(V2-B)
      F2= X2*DLOG(V3-B)-X3*DLOG(V2-B)
      F3= DLOG(V3-B)-DLOG(V2-B)
      F4= X2*DLOG(V1-B)-X1*DLOG(V2-B)

      FUNC2= F1*F2-F3*F4

      RETURN
      END
```

Y.   Function FSIN

```
      FUNCTION FSIN(A)
      IMPLICIT DOUBLE PRECISION (A-Z)
      IF (A.LT.0.) THEN
        X= -1.*DSIN(DABS(A))
      ELSE
        X= SIN(A)
      END IF
      FSIN= X
      RETURN
      END
```

REFERENCES

1. Hastings, L.J., and Schmidt, G.R.: "The Marshall Space Flight Center Cryogenic Fluid Management Program." AIAA Space Programs and Technologies Conference, AIAA Paper 93-4224, September 21–23, 1993.

2. Schmidt, G.R., and Hastings, L.J.: "Cryogenic Fluid Management Program at MSFC." AIAA Space Programs and Technologies Conference, AIAA Paper 90-3711, September 19–21, 1990.

3. Blatt, M.H., Stark, J.A., and Siden, L.E.: "Low Gravity Propellant Control Using Capillary Devices in Large Cryogenic Vehicles—Design Handbook." General Dynamics Convair Division, Report GDC-DDB70-006, Contract NAS8-21465, August 1970.

4. Dodge, F.T.: "Fluid Management in Low Gravity." Low-Gravity Fluid Dynamics and Transport Phenomena, American Institute of Aeronautics and Astronautics, Washington, DC, 1990, pp. 1–18.

5. Symons, E.P.: "Wicking of Liquids in Screens." NASA Technical Note D-7657, May 1974.

6. Burge, G.W., and Blackmon, J.B.: "Study and Design of Cryogenic Propellant Acquisition Systems—Vol. II." McDonnell Douglas Astronautics Co., Report MDC G5038, Contract NAS8-27685, December 1973.

7. Blackmon, J.B.: "Design, Fabrication, Assembly, and Test of a Liquid Hydrogen Acquisition Subsystem." McDonnell Douglas Astronautics Co., Report MDC G5360, Contract NAS8-27571, May 1974.

8. Cady, E.C.: "Design and Evaluation of Thermodynamic Vent/Screen Baffle Cryogenic Storage System." McDonnell Douglas Astronautics Co., Report MDC G5360, NASA CR-134810, June 1975.

9. Paynter, H.L., Page, G.R.: "Acquisition/Expulsion Orbital Propulsion System Study—Vol. II: Cryogenic Design." NASA CR-134154, October 1973.

10. Warren, R.P.: "Acquisition System Environmental Effects Study." Martin Marietta Corp., Report MCR-75-21, Contract NAS8-30592, May 1975.

11. Warren, R.P.: "Measurements of Capillary System Degradation." AIAA Paper 75-1197, September 1975.

12. Bennett, F.O.: "Design and Demonstrate the Performance of Cryogenic Components Representative of Space Vehicles—Start Basket Liquid Acquisition Device Performance Analysis." General Dynamics Space Systems Division, Report GDSS-CRAD-87-004, Contract NAS8-31778, February 1987.

13. Meserole, J.S., and Jones, O.S.: "Pressurant Effects on Cryogenic Liquid Acquisition Devices." AIAA Journal of Spacecraft and Rocket, vol. 30, No. 2, 1993, pp. 236–243.

250

14. Burelback, J.P., Bankoff, S.G., and Davis, S.H.: "Nonlinear Stability of Evaporating/Condensing Liquid Films." Journal of Fluid Mechanics, vol. 195, October 1988, pp. 463–494.

15. Luikov, A.V., and Vasiliev, L.L.: "Progress in Heat Pipe and Porous Heat Exchanger Technology." International Journal of Heat and Mass Transfer, vol. 18, No. 2, 1975, pp. 177–190.

16. Ostrach, S., and Kamotani, Y.: "Recent Developments in Oscillatory Thermocapillary Flows." Proceedings of AIAA/IKI Microgravity Science Symposium, American Institute of Aeronautics and Astronautics, Washington, DC, 1992, pp. 25–32.

17. Ostrach, S.: "Fluid Mechanics in Crystal Growth—The 1982 Freeman Scholar Lecture." ASME Journal of Fluids Engineering, vol. 105, No. 1, 1983, pp. 5–20.

18. Fischat, G.H., Herr, K., and Barklage-Hilgefort, H.: "Probleme bie der Vorbereitung Glastechnischer Untersuchungen in Weltraum." Glastechnische Berichte, vol. 53, No. 1, 1980, pp. 1–9.

19. Kou, S., and Sun, D.: "Fluid Flow and Weld Penetration in Stationary Arc Welds." Metallurgical Transactions, vol. 16, No. 2, 1985, pp. 203–213.

20. Bergman, T.L., and Ramadhyani, S.: "Combined Buoyancy- and Thermocapillary-Driven Convection in Open Square Cavities." Numerical Heat Transfer, vol. 9, No. 4, 1986, pp. 441–451.

21. Bergman, T.L., and Keller, J.R.: "Combined Buoyancy, Surface Tension Flow in Liquid Metals." Numerical Heat Transfer, vol. 13, No. 1, 1988, pp. 49–63.

22. Jue, T.C., Ramaswamy, B., and Akin, J.E.: "Computation of Thermocapillary and Buoyancy Affected Cavity Flow Using Semi-Implicit FEM." Numerical Methods in Thermal Problems, vol. 7, Part 1, 1991, pp. 402–312.

23. Hadid, H., and Roux, B.: "Buoyancy- and Thermocapillary-Driven Flows in Differentially Heated Cavities for Low Prandtl Number Fluids." Journal of Fluid Mechanics, vol. 235, February 1992, pp. 1–36.

24. Sen, A.K., and Davis, S.H.: "Steady Thermocapillary Flows in Two-Dimensional Slots." Journal of Fluid Mechanics, vol. 121, August 1982, pp. 163–186.

25. Sen, A.K.: "Thermocapillary Convection in a Rectangular Cavity With a Deformable Interface." Physics of Fluids A, vol. 29, No. 11, 1986, pp. 3881–3883.

26. Strani, M., Piva, R., and Graziani, G.: "Thermocapillary Convection in a Rectangular Cavity: Asymptotic Theory and Numerical Simulation." Journal of Fluid Mechanics, vol. 130, May 1983, pp. 347–376.

27. Zebib, A., Homsy, G.M., and Meiburg, E.: "High Marangoni Number Convection in a Square Cavity." Physics of Fluids A, vol. 28, No. 12, 1985, pp. 3467–3476.

28. Carpenter, B., and Homsy, G.M.: "Combined Buoyant-Thermocapillary Flow in a Cavity." Journal of Fluid Mechanics, vol. 207, October 1989, pp. 121–132.

29. Cuvelier, C., and Driessen, J.M.: "Thermocapillary Free Boundaries in Crystal Growth." Journal of Fluid Mechanics, vol. 169, August 1986, pp. 1–26.

30. Fu, B.-I., and Ostrach, S.: "Numerical Solutions of Thermocapillary Flows in Floating Zones." Transport Phenomena in Materials Processing, American Society of Mechanical Engineers, New York, NY, 1983, pp. 1–9.

31. Shen, Y., Neitzel, P., Jankowski, D., and Mittelmann, H.: "Energy Stability of Thermocapillary Convection in a Model of the Float-Zone, Crystal Growth Process." Journal of Fluid Mechanics, vol. 217, August 1990, pp. 639–660.

32. Kobayashi, N.: "Steady Convection Caused by the Temperature Inhomogeneity in a Cylindrical Floating Zone." Japanese Journal of Applied Physics, vol. 27, No. 1, 1988, pp. 20–24.

33. Xu, J.-J., and Davis, S.H.: "Instability of Capillary Jets With Thermocapillarity." Journal of Fluid Mechanics, vol. 161, December 1985, pp. 1–25.

34. Durancea, J.L., and Brown, R.A.: "Finite Element Analysis of Melt Convection and Interface Morphology in Earthbound and Microgravity Floating Zones." Drops and Bubbles Third International Colloquium, American Institute of Physics, New York, NY, 1989, pp. 133–144.

35. Hyer, J., Jankowski, D., and Neitzel, G.: "Thermocapillary Convection in a Model Float Zone." AIAA Journal of Thermophysics and Heat Transfer, vol. 5, No. 4, 1991, pp. 577–582.

36. Lan, C.W., and Kou, S.: "Heat Transfer, Fluid Flow, and Interface Shapes in Floating Zone Crystal Growth." Journal of Crystal Growth, vol. 108, 1991, pp. 351–366.

37. Zhang, Y., and Alexander, J.I.D.: "Surface Tension and Buoyancy-Driven Flow in a Nonisothermal Liquid Bridge." International Journal for Numerical Methods in Fluids, vol. 14, No. 2, 1992, pp. 197–215.

38. Kamotani, Y., and Platt, J.: "Effect of Free Surface Shape on Combined Thermocapillary and Natural Convection." AIAA Journal of Thermophysics and Heat Transfer, vol. 6, No. 4, 1992, pp. 721–726.

39. Potash, M., and Wayner, P.: "Evaporation From a Two-Dimensional Extended Meniscus." International Journal of Heat and Mass Transfer, vol. 15, No. 10, 1972, pp. 1851–1863.

40. Renk, F.J., and Wayner, P.C.: "An Evaporating Ethanol Meniscus—Part II: Analytical Studies." ASME Journal of Heat Transfer, vol. 101, No. 1, 1979, pp. 59–62.

41. Werhle, V., and Voulelikas, G.: "Evaporation From a Two-Dimensional Meniscus." American Institute of Aeronautics and Astronautics Journal, vol. 23, No. 2, 1985, pp. 309–313.

42. Mirzamoghadam, A., and Catton, I.: "A Physical Model of the Evaporating Meniscus." ASME Journal of Heat Transfer, vol. 110, No. 1, 1988, pp. 201–207.

43. Swanson, L.W., and Herdt, G.C.: "Model of the Evaporating Meniscus in a Capillary Tube." ASME Journal of Heat Transfer, vol. 114, No. 2, 1992, pp. 434–441.

44. Swanson, L.W., and Peterson, G.P.: "The Evaporating Extended Meniscus in a V-Shaped Channel." AIAA Journal of Thermophysics and Heat Transfer, accepted for publication in 1993.

45. Chen, H., Oshima, K., and Hinada, M.: "Numerical Analysis of Thermocapillary and Evaporating Flows at Low Bond Number." Proceedings of the Symposium on Mechanics of Space Flight, Sagamihara, Japan, November 24–25, 1988, pp. 39–53.

46. Chung, T.J.: "Continuum Mechanics." First edition, Prentice Hall, Englewood Cliffs, NJ, 1988.

47. Slattery, J.C.: "Momentum, Energy, and Mass Transfer in Continua." Second edition, Krieger Publishing, Huntington, NY, 1981.

48. Slattery, J.C.: "Interfacial Transport Phenomena." First edition, Springer-Verlag, New York, NY, 1990.

49. Kennard, E.: "Kinetic Theory of Gases." First edition, McGraw-Hill, New York, NY, 1938.

50. Patankar, S.V.: "Numerical Heat Transfer and Fluid Flow." Hemisphere, Washington, DC, 1980.

51. Chung, T.J.: "Finite Element Analysis in Fluid Dynamics." McGraw-Hill, New York, NY, 1978.

52. Concus, P.: "Static Menisci in a Vertical Right Circular Cylinder." Journal of Fluid Mechanics, vol. 34, No. 3, 1968, pp. 481–495.

53. Geiger, F.: "Hydrostatics of a Fluid in a Cylindrical Tank at Low Bond Numbers." Brown Engineering Research Laboratories, Technical Note R-207, July 1966.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE February 1994	3. REPORT TYPE AND DATES COVERED Technical Paper

4. TITLE AND SUBTITLE
Thermocapillary Flow With Evaporation and Condensation and Its Effect on Liquid Retention in Low-G Fluid Acquisition Devices .
MSFC Center Director's Discretionary Fund Final Report. Project No. 91-15

5. FUNDING NUMBERS

6. AUTHOR(S)

George R. Schmidt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

8. PERFORMING ORGANIZATION REPORT NUMBER

M-743

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TP-3463

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified—Unlimited
Subject Category: 34

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The steady motion, thermal and free surface behavior of a volatile, wetting liquid in microgravity are studied using scaling and numerical techniques. The objective is to determine whether the thermocapillary and two-phase convection arising from thermodynamic nonequilibrium along the porous surfaces of spacecraft liquid acquisition devices could cause the retention failures observed with liquid hydrogen and heated vapor pressurant. The study also examines why these devices seem immune to retention loss when pressurized with heated helium or heated directly through the porous structure. Results show that highly wetting fluids exhibit large negative and positive dynamic pressure gradients towards the meniscus interline when superheated and subcooled, respectively. With superheating, the pressure variation and recoil force arising from liquid/vapor phase change exert the same influence on surface morphology and promote retention. With subcooling, however, the pressure distribution produces a suction that degrades mechanical equilibrium of the surface. This result indicates that thermocapillary-induced deformation arising from subcooling and condensation is the likely cause for retention loss. In addition, increasing the level of nonequilibrium by reducing accommodation coefficient suppresses deformation and explains why this failure mode does not occur in instances of direct screen heating or pressurization with a heated inert gas.

14. SUBJECT TERMS
microgravity, surface tension, liquid surfaces, two-phase flow, thermocapillary flow, evaporation, condensation, liquid acquisition device bubble-point pressure, finite element, computational fluids

15. NUMBER OF PAGES
270

16. PRICE CODE
A12

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited